Free Inside Your Robot

BRITAIN'S FIRST ROBOTICS MAGAZINE

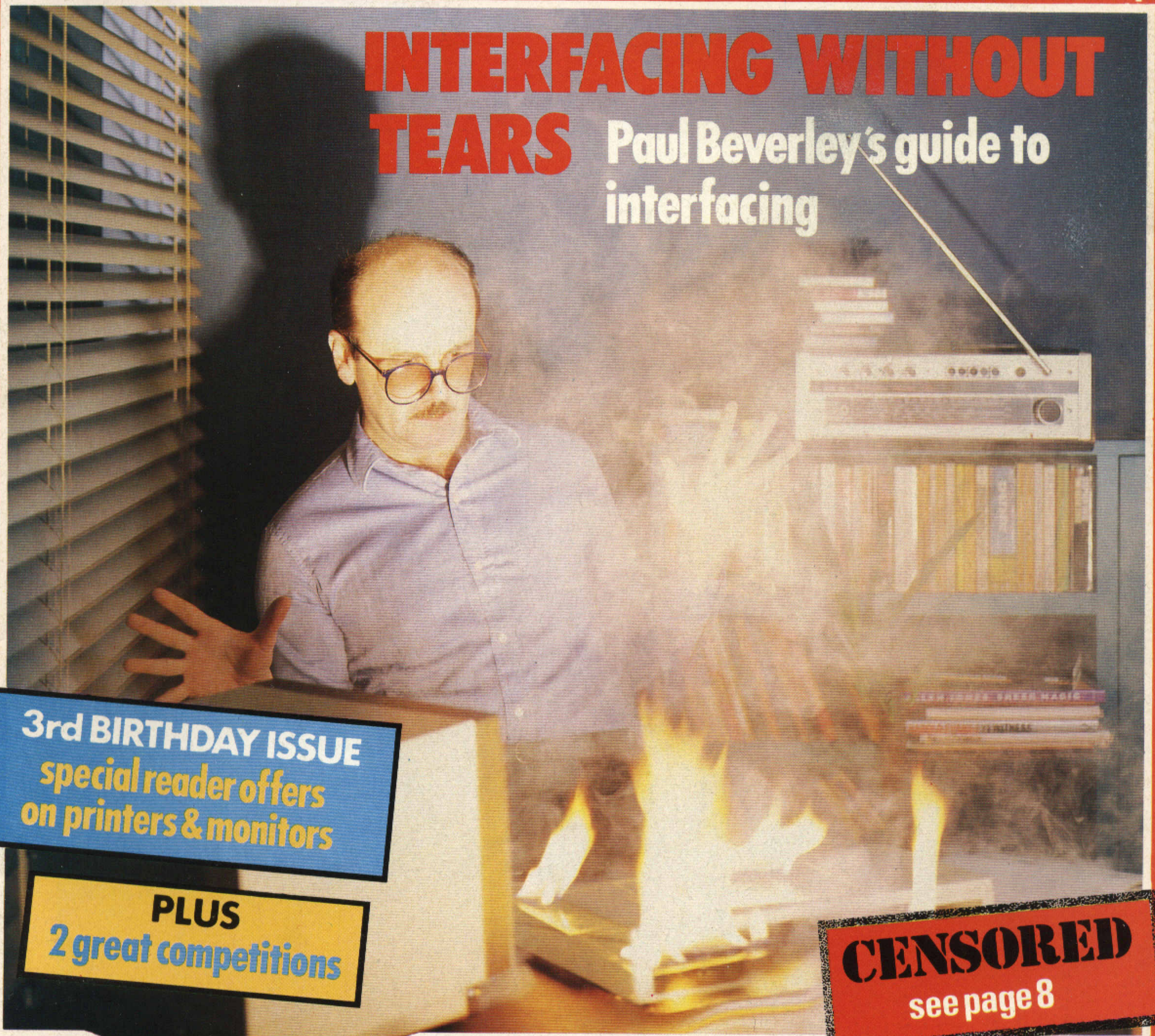
TH... COMPUTER PROJECTS MAGAZINE JUNE 1984

...ECTRONICS & COMPUTING MONTHLY

8 MEMO NOTE 14 DATA NWK 20 26 MICRO GRAPHIC 31 ROBOT 35 ROBOT BOOK 41 2 ROBOT GRIL 53 REF TOOL STAR 64 67 WORD PROC BOOK REVIEW

AN EMAP PUBLICATION

USA $2.95 Germany D5.80 90p



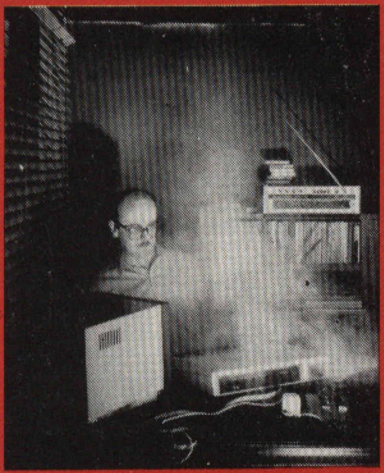
BYTES BY THE MAINS – PLUG IN TO A READY-MADE NETWORK

BUYERS GUIDE TO UTILITY SOFTWARE

DRAGON EPROM PROGRAMMER • SPECTRUM MODS

ELECTRONICS & COMPUTING Contents

Vol. 4 Issue 6



COVER PHOTO BY ROB BRIMSON

Electronics & Computing Monthly
Scriptor Court, 155 Farringdon Road,
London, EC1R 3AD

Editorial 01-833-0846
Editor Gary Evans
Assistant Editor William Owen
Production Editor Liz Gregory
Administration Serena Hadley

Advertising 01-833-0531
Advertisement Manager Anthony Herman
Chief Executives Richard Jansz
Francis Lee
Classified Martin Derx

Production 01-833-0531
Art Editor Jeremy Webb
Make-up Time Graphics
Publisher Alfred Rolington

Distribution
EMAP National Publications

Published by
EMAP Business and
Computer Publications

Printed by
Riverside Press, England

Subscriptions
Electronics & Computing Monthly,
(Subscription Department),
Competition House, Farndon Road,
Market Harborough, Leicestershire.

Electronics & Computing Monthly is normally published on the 13th day of each month.

ABC

PROJECTS

FEATURES

REVIEWS

PLUS

EDITORIAL EDITORIAL EDITORIAL

Censored!

Last month we published the first part of our Memex project. This was a design that allowed an extra 20K of RAM to be added to the 32K featured as standard on the BBC micro. Shortly after the issue had appeared on the bookstalls we received a letter from a firm of solicitors acting on behalf of a company that were marketing a product offering a similar specification to our project. The letter stated that a patent in respect of the commercial product had been applied for and that our design was infringing that patent. The letter went further and in effect suggested that the authors of the project had copied the design of the commercial project. This is something that both the authors of Memex and *E&CM* strongly deny and we are at present setting out to establish this to the satisfaction of all concerned.

It is unfortunate, however, that due to the rather slow processes by which disputes of this nature are resolved we are unable to publish the second part of the project in this issue. It is to be hoped that we can sort things out over the next few weeks and bring you part two in our next issue. Stay tuned for further news and more thoughts upon this sort of legal action in the July issue of *E&CM*.

Place an order

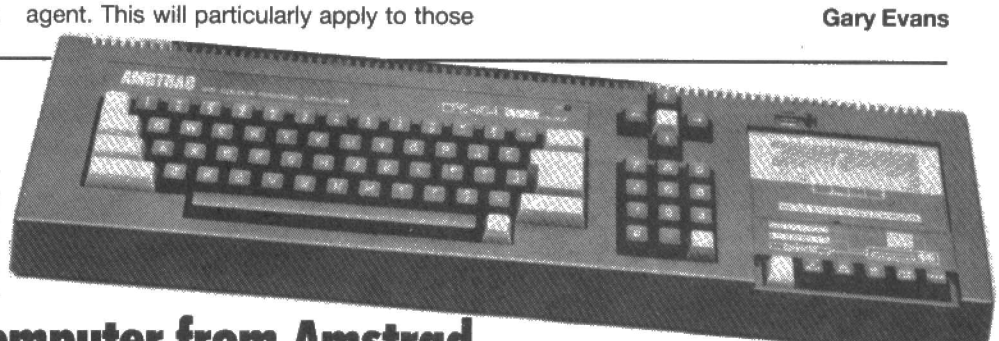
As from next month, because of a change in the way in which *E&CM* is distributed, you may find it more difficult than usual to get hold of your copy from the local newsagent. This will particularly apply to those of you that usually buy *E&CM* from the small newsagent 'round the corner'. The solution to this problem is to place a regular order with your local shop. In this way a copy will be reserved for you and there will be no danger of missing a single issue of *E&CM*.

An alternative is to place an annual subscription to the magazine. This will ensure that a copy will land on your doorstep each and every month saving you the trouble of the trek down to the newsagent.

Either way we make the task as easy as possible by providing the appropriate form. In the case of those of you wishing to place an order with your newsagent, turn to page 39, and for those wanting a subscription, the form is on page 43.

Gary Evans

CPC464 revisited



the low cost colour computer from Amstrad.

Our exclusive preview of the new Amstrad CPC464 colour computer pre-empted the blanket coverage that the machine has received in the press during the past few weeks. The verdict of the numerous assessments of the 464 has been unanimous, with everybody agreeing with the sentiments of our report namely that the 464 will cause one of the biggest shake-ups in the market since the launch of the Spectrum. Indeed some reviewers are suggesting that the days of the Atmos, Dragon, Spectrum, CBM64, and even the BBC micro, are numbered.

The price and performance of the 464 would certainly suggest that the competition is going to have to do something if the Amstrad computer is not to corner the sub-£500 micro market as we move into the latter half of this year. The action that will seem most attractive to some of the above named companies will be a price cutting exercise. The 464 offers such good value for money however that any price cuts will have to be quite savage. In some cases the margins just will not be there.

Given that the options open to the other manufacturers are thus limited that the Amstrad organisation are undoubtedly extremely capable when it comes to shifting products if Amstrad can land enough machines in the run up to Christmas the CPC464 could make a clean sweep of the market.

Even faster

It is in the nature of reviews like last month's preview that some of the details may not be spot on. In the main the article got things right but the benchmark timings were based on a pre-production model that did not have the final version of the custom ULA fitted. The corrected timings reproduced here show that the 464 is in all cases slightly faster than the figures of last month suggest. The prices quoted last month were also some £20 out and the full price list for the four systems, this time including VAT is opposite.

The disk based systems mentioned in the price list will not be available until September and many people wanting to make use of the 464 in a business environment may well decide to wait until then before deciding whether or not to buy the machine. The disk system was demonstrated at the official launch of the computer however where it was shown in conjunction with the CP/M based word processor, Wordstar.

Full review

Without getting our hands on the computer for a full test there is little we can add to our report of last month. At present all that we, and most other magazines, have managed is a day's hands-on at Amsofts Brentwood HQ. We have been promised a machine in the near future and when we get our hands on one we'll add a bit more flesh to the review as well as bringing you a couple of add-on projects, most notably an A/D converter and a serial interface.

Corrected Benchmarks

BM1	1.09
BM2	3.28
BM3	9.16
BM4	9.61
BM5	10.20
BM6	19.03
BM7	30.18
BM8	34.20

Price List

CPC464 +green monitor	**£229**
CPC464 + RGB monitor	**£329**
CPC464 + disk drive + green monitor	**£429**
CPC464 + disk drive + RGB monitor	**£529**
All prices include VAT	

NEWS NEWS NEWS NEWS

More QL News . . . Yawn

Sorry to bore you, but just a few more words on the latest million seller soap opera, the QL saga.

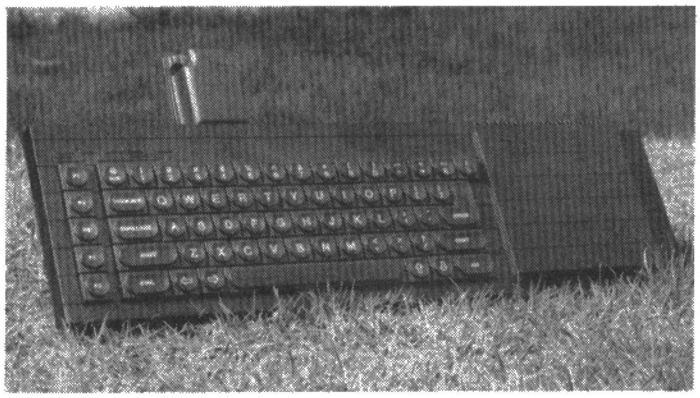
By the time this magazine hits the streets, the first few Sinclair QLs will be dropping through the nation's letter boxes, contrary to the dire predictions of hard-bitten cynical computer journalists that the QL would be the greatest hoax since Piltdown Man.

However, this does not mean to say the QL isn't a botch job. As you will read in our short review on page 38, the operating system, QDOS required an extra 16K ROM, and Sinclair's temporary solution has been to stick an EPROM in the ROM cartridge slot at the back of the machine. It doesn't look very nice; it isn't very elegant.

Bill Nichols, Sinclair PR man, says that the EPROM OS will be included only during the first two months of despatch, and early buyers will get an upgrade when the full 48K ROMs are available (but will probably have to send their machines back to Sinclair for the ammendments).

However, the first batch of machines will not have full BASIC editing facilities: Sinclair botched that as well. For example any line written in with an error cannot be recalled by the EDIT command – the whole line must be written in again with the correction. Only lines without syntax errors can be edited. An upgrade, rumour has it, will not be available on the BASIC. It is also likely that later machines will have a number of new SuperBASIC commands to take advantage of the bigger OS/BASIC ROM.

PS. The consolation prize for waiting up to four months for a QL is a free RS232 cable and connector from Sinclair.



What price quality?

The British software industry (read home computer software industry) is still going through an agony of birth pains.

The preoccupation of games houses with pirates is rapidly being replaced by pricing paranoia. Imagine, the Liverpool based software company who are well known for their constant missives to the press about piracy, recently announced sweeping price reductions to all their games. This move provoked an immediate outcry from competitors and Imagine, either because of pressure or a rethinking of budgets, have recanted – prices are back to normal.

The National Micro Centre, a rapidly growing chain of retail computer outlets, went so far as to issue a statement publicly condemning Imagine's price cuts. They say that a price war 'would result in Britain's world-beating software industry sliding into mediocrity'.

It should not be forgotten that NMC have an interest in keeping software prices – and profits – as high as they can. The pressure for high prices contradicts the industry's demand for action against the pirates, who thrive when the punters can't afford the genuine goods. Imagine, had they gone through with their proposals, should have been congratulated. The Liverpool company has either succumbed to pressure or the temptation of a cheap publicity stunt.

Start of Term at Swansea University

The BBC micro's capabilities as a terminal are almost as renowned as its computing powers, and because of the interest in using the machine as a mainframe terminal/work station Swansea University set out to develop suitable emulation software.

They have come up with an EPROM called Uniterm. It is priced at £40 in small quantities, and features include a full menu driven system; graphics mode with alpha, vector, point and graphics cursor modes, screen dumps, full VT52 emulation including printer options, and teletext mode to full Prestel standard.

The system is RS232 based, and is called up instantly by typing *TERM. Control can be passed between Uniterm and BASIC without disturbing parameters and variables held in either.

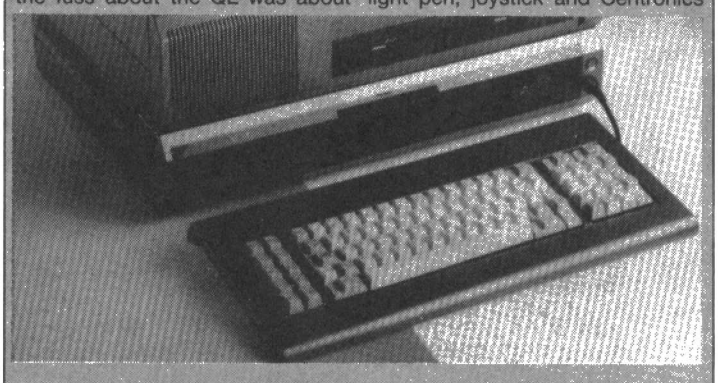
Uniterm has a graphics mode (Tektronix T4014 compatible). In cursor mode the editing keys control a full screen crosshair providing 'interactive digitisation', with up to four colours available at any one time.

Uniterm can be obtained from Dr. G. J. Morris, University College Swansea, Singleton Park, Swansea SA2 8PP.

An Advance in computing

Of the many new computers in the £400 range to come on the market in recent months, only two – the QL and the Advance – have used the 16-bit option.

The Advance is a British built computer with excellent specifications; in fact one wonders what all the fuss about the QL was about when the two machines stand side by side.



There are two versions: the standard Model A, and the Model B which is aimed at the business market with two 5¼" disc drives and a suite of business software.

The Model A is intended as a computer for the home. The main processor is a (true) 16-bit Intel 8086 which steams along at 4.77MHz. User RAM is 128K expandable to 256K, plus an extra 16K dedicated to driving video signals. ROM is 64K and includes diagnostics, BASIC, and cassette operating system.

The keyboard is separated from the processor and looks to be superior to that of the QL. It has 84 keys, including ten function keys and numeric pad, all with full travel and tactile response.

Graphics resolution is 320 x 200 or 640 x 200 with a maximum of 16 colours, scroll and reverse image. The text display is either 80 or 40 column.

Interfaces include cassette port, light pen, joystick and Centronics parallel interface. The Model A can of course be upgraded using the Model B disc drive expansion unit.

Both A and B models are software and hardware compatible with the IBM PC (but the drives use MS/DOS, not PC/DOS).

At £404.99 this range of features constitutes a remarkable achievement. Obviously the Advance will not be competing with the likes of Amstrad and Tatung, and unlike the QL, the machine will need full upgrading to the Model B standard before it becomes a useable business system. However for the serious amateur programmer the Advance must be an attractive machine. *E&CM* hopes to carry a full review of the Advance in the July issue.

Get Alert . . . stay in touch

Discovering articles in the computer press on a particular machine, software package, games listing etc. is now a lot easier thanks to a very useful index of computing articles compiled by Information Alert.

The pamphlet, called Microcomputer Alert, is updated every two months and contains full details of all articles published in a selection of 20 magazines and journals, including *E&CM* of course.

Microcomputer Alert can be obtained from Information Alert, 38 Part Street, Southport PR8 1HY for a yearly subscription of £25. A sample copy costs £2.

BYTES by the MAINS

Robert Penfold describes a pair of units that allow your home's ring main to be transformed into a ready made comms network.

Traditionally there are only three ways of transmitting data from one computer to another. For short distances a cable connection from one computer to the other is used, or for long distances a radio link or the normal telephone system (via a couple of modems) are utilized. This novel design is an alternative method of short distance communication which makes use of the mains wiring to provide the link between the two computers, and in its method of operation has similarities with radio and telephone data links. The use of the mains wiring eliminates the need to install a cable running between the two computers (or whatever items of computer equipment are involved), and does in fact render any special installation work unnecessary. It also means that the data link is easily moved from one room to another along with the other computer equipment, and rerouting of connection cables is avoided.

The transmitter has an input that will operate with RS232C, RS423, or standard 5 volt logic input signals. The receiver provides an output at RS232C levels (nominally plus and minus 8 volts), but these could obviously be converted to 5 volt logic levels without difficulty if necessary. The unit has been tested using baud rates of up to 1200, but it should operate at somewhat higher baud rates if necessary.

System operation

Most data links rely on a system known as frequency shift keying, or FSK as it is known in its abbreviated form. This design is no exception, and the block diagram of the system appears in **Figure 1.**

With a frequency shift keying link the transmitter provides two output frequencies; one when the input signal is high and the other when it is low. The transmitter is therefore basically just a VCO with its control input driven by the input signal. In this case the VCO has its input driven via an inverter (which is needed to ensure that the signal does not undergo a phase inversion through the system), and an electronic switch. When switched off the latter allows the VCO to operate at its normal frequency, but it pulls it to a higher frequency when it is switched on. This system ensures that the two output frequencies of the VCO are independent of the two input signal voltages, which in practice will vary considerably from one piece of equipment to another. A transformer couples the high frequency (about 200kHZ) output signal into the mains "N" lead, which, together with the mains "E" lead, provides the link between the transmitter and receiver.

The receiver has a transformer at its input, and this provides two main functions. Firstly it steps up the low impedance, low level input signal to a higher voltage higher impedance signal that is a better match for the next stage of the unit. The transformer is a tuned type which provides a bandpass filter action, and helps to cut down the inevitable out-of-band signals present on the input signal. Another function of the transformers at both ends of the system is to give isolation from the mains supply.

A phase locked loop is used to decode the incoming FSK signal, and this circuit is comprised of the next four stages of the unit. A phase comparator is used to compare the received tones with the output of a VCO. The output of the phase comparator is amplified, smoothed by a lowpass filter, and then used as the control voltage for the VCO. The circuit is arranged so that this gives a high control voltage for the VCO if it is at a lower frequency than the received tone, or lagging in phase in comparison to this signal. This high control voltage raises the frequency of the VCO. Conversely, a low control voltage is produced if the VCO is at a higher frequency than the received tone or is leading it in phase. This gives reduced VCO operating frequency.

The overall effect of this is to lock the VCO onto the same frequency as, and in-phase with, the incoming tones. In this application it is not the output of the VCO that is of interest, but the control voltage. This rises and falls in sympathy with the received frequency, and in this case the two onput tones produce two different voltages at the control input of the VCO. These are fed to a voltage comparator where they are compared to a reference voltage which is roughly half way between the two VCO control voltages. The two input voltages therefore trigger the output of the comparator fully high and low, and as the circuit is powered from a dual balanced supply this gives the required RS232C output levels without any further interfacing.

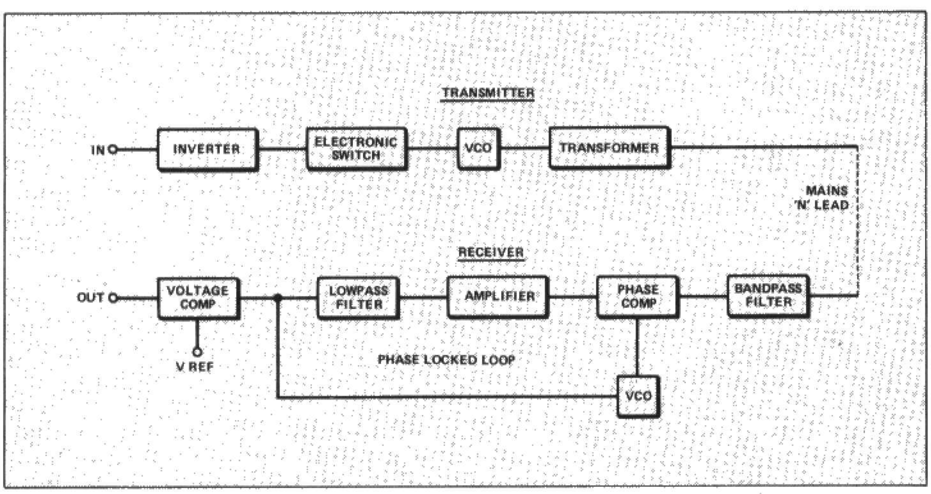


Figure 1. Block diagram of the system.

Transmitter circuit

The full circuit diagram of the transmitter is shown in **Figure 2.** The oscillator is a conventional 555 astable circuit having its output coupled to the "N" side of the mains supply via C2, T1, and C4. The operating frequency of the oscillator can be adjusted by means of VR1, and this is adjusted to bring the transmission frequencies to suitable figures for the receiver (which does not have adjustable tuning).

Q1 is used in the inverter stage, and Q2 is the electronic switch. With Q2 switched off IC1 functions normally, with C1 first charging to two thirds of the supply voltage through R4, R5, and VR1, and then discharging to one third of the supply voltage by way of VR1, R5, and an internal transistor of IC1. When Q2 is switched on, R6 is shunted across the internal potential divider of IC1 which sets the two thirds of V+ discharge potential, and this voltage is consequently reduced. As C1 now has to charge to and discharge from a lower potential, both these times are shortened and the output frequency is boosted.

The power supply is a simple non-stabilised type using T1 to provide mains isolation and a voltage step-down, and D1 plus D2 to provide fullwave push-pull rectification. The loaded output voltage across smoothing capacitor C5 is approximately 8 volts. Although good frequency stability from the oscillator is important, it is unnecessary to use a stabilised supply as the 555 has an output frequency which is almost totally independent of the supply voltage.

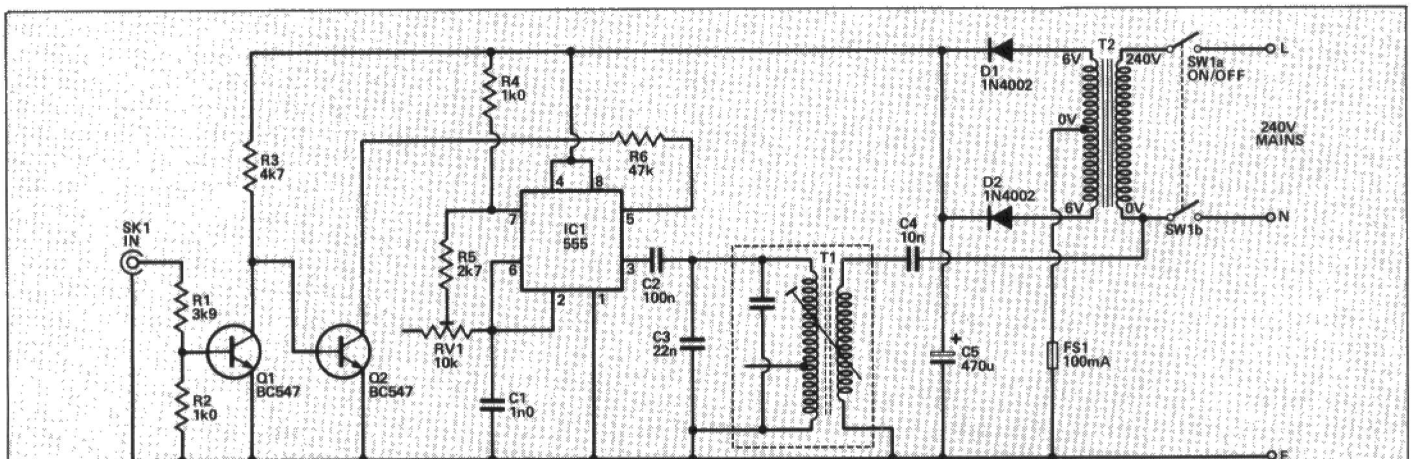
Receiver circuit

Due to the use of a phase locked looped integrated circuit which is specifically designed for use in FSK applications, the receiver circuit is surprisingly simple. The full circuit diagram of the receiver is shown in **Figure 3.**

The phase locked loop device is an NE565 (IC3). This has two inputs, but in this application it is only the input at pin 2 which is used. Both inputs must be biased to the 0 volt supply rail, and R12 plus R13 provide this biasing. The input signal is coupled to the primary winding of T3 by C14, and C11 couples the output from the secondary of T3 to the input of IC3. T3 is actually an ordinary 455kHZ last IF transformer which is used in reverse in this application, but it nevertheless operates efficiently. C12 is used to shunt the internal tuning capacitor of T3 and reduce the resonant frequency to the desired figure (the same technique is used with T1 in the receiver incidentally).

Pins 4 and 5 of IC3 are the VC0 output and phase comparator input, and in this application they are simply connected together. R11 and C10 are the timing components for the VC0, and C9 is needed to prevent instability. C8, in conjuction with an internal resistor of IC3, acts as the lowpass filter. R8, R9, C6 and C7 provide additional smoothing on the output to the comparator. Pin 6 of IC3 provides the reference voltage for the other input of the comparator. The comparator is actually just an ordinary 741C operational amplifier (IC2), and this provides perfectly adequate performance for this application.

The power supply has to be a dual balanced type since both IC2 and IC3 require dual supplies. The power supply is again a simple non-stabilised type, and it differs from the one used in the transmitter only in that an additional rectifier and smoothing circuit are included to generate the negative supply. The loaded supply potentials are approximately plus and



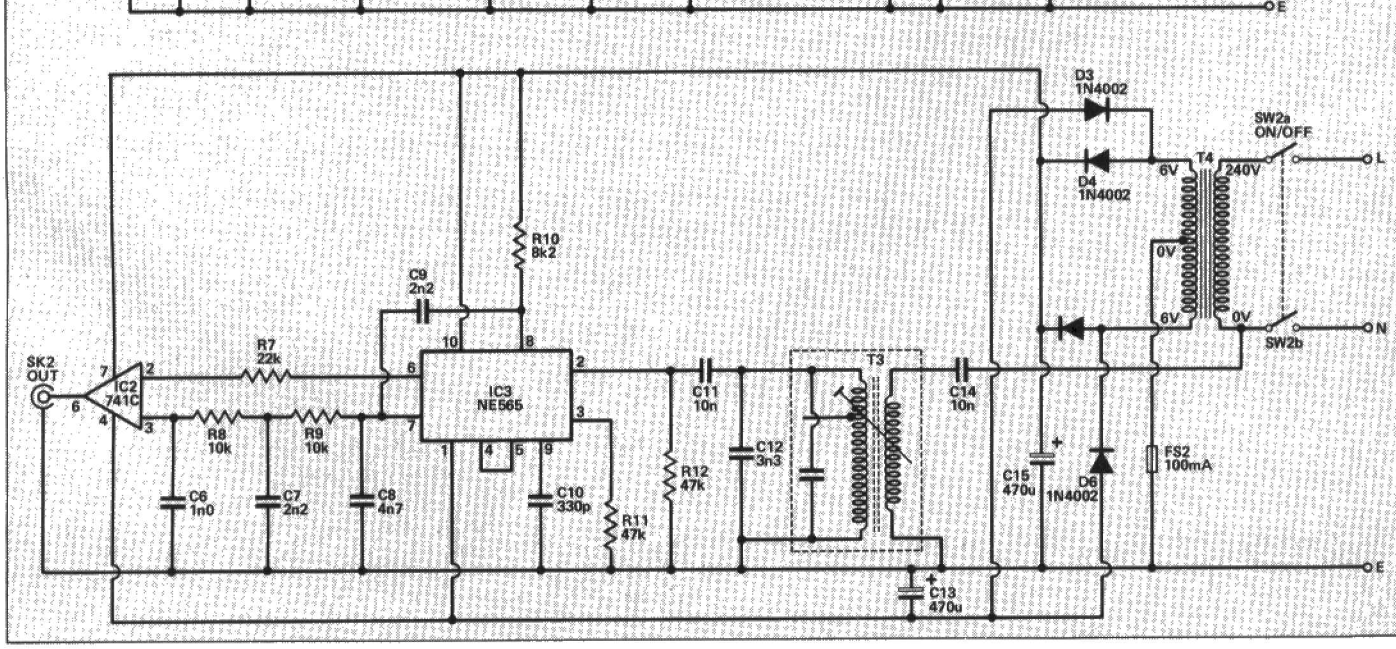
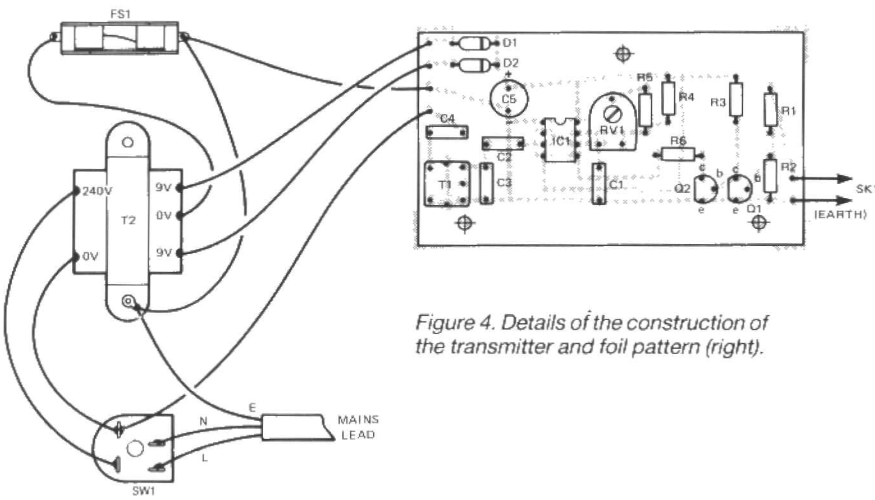


Figure 2 (top) Circuit diagram of the transmitter and Figure 3 (bottom) shows the circuit diagram for the receiver.



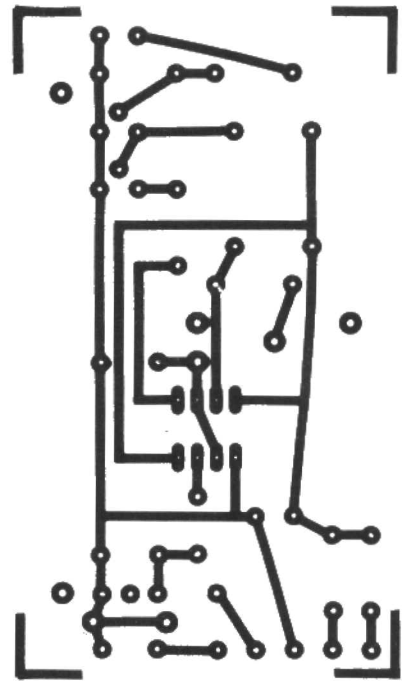
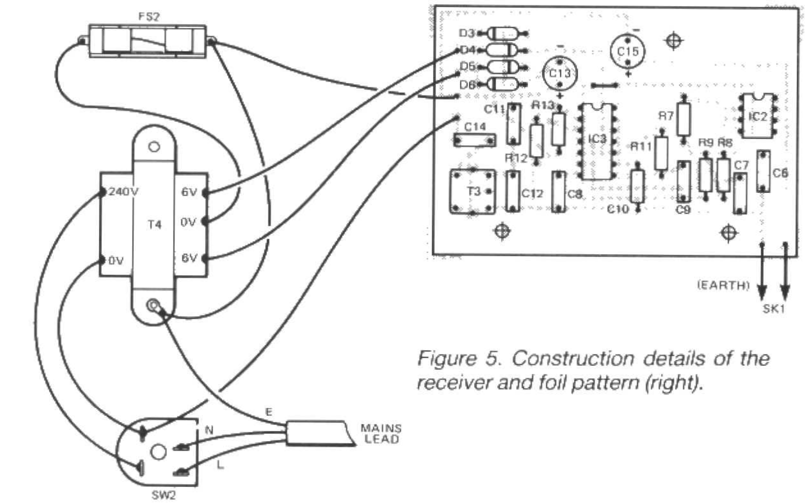


Figure 4. Details of the construction of the transmitter and foil pattern (right).



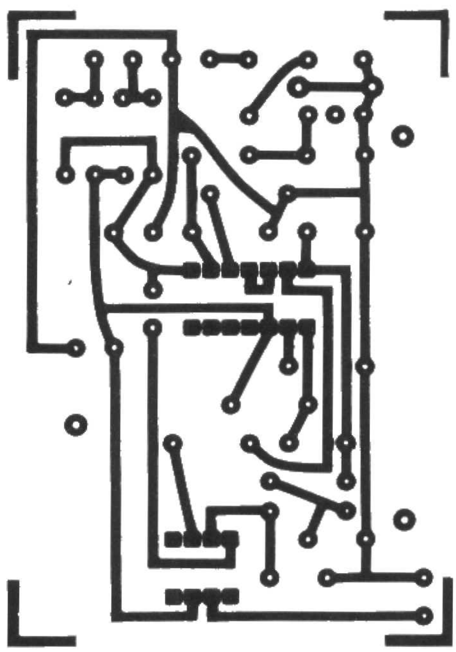


Figure 5. Construction details of the receiver and foil pattern (right).

minus 8 volts, which is more than adequate to provide RS232C output levels where a minimum of plus and minus 3 volts is required. The operating frequency of the VCO in the NE565 is not significantly affected by variations in supply voltage, and a non-stabilised supply is perfectly satisfactory.

Construction

The transmitter and receiver units are constructed very much along the same lines, and a metal instrument case having approximate outside dimensions of 150 by 100 by 50 millimetres is needed for each of them. As can be seen from the photographs, the front panel layout is very simple with the on/off switch towards the left and the input or output socket at the opposite end of the panel. 3.5 millimetre jacks are used as the input and output sockets on the prototypes, but any desired connectors having two or more ways can be utilized here. A hole for the mains lead is drilled in each rear panel, roughly opposite the on/off switch, and these holes should be fitted with grommets to protect the cables.

The mains transformer and fuseholder for each unit are mounted on the chassis, well towards the left hand side of the unit so that there is sufficient space for the relevant printed circuit board on the right hand section of the chassis. A soldertag is fitted on one of each mains transformer's mounting screws, and these provide convenient chassis connection points. Note that the fuses must be of the anti-surge variety, and not the more usual quick-blow type (which, due to the initial charge current of the smoothing capacitors, would live up to their name).

Details of the transmitter printed circuit board are shown in **Figure 4.** Provided the specified types of component are used (particularly the capacitors) there should be no difficulty in constructing the board. Fit Veropins at the points where connections to the off-board components will be made. The completed board is mounted on the chassis using 6BA or M3 mountings, and these must include spacers about 6 or 12 millimetres long to keep the connections on the underside of the board clear of the aluminium chassis.

The hard-wiring is then added, and details of this are included in **Figure 4.** This is again quite straightforward, but take great care with the mains wiring. In particular make sure that the lead which connects to C4 on the printer circuit board comes from the "N" and not the "L" side of the mains supply (via S1). Thoroughly check the finished unit for errors before connecting it to the mains supply.

Construction of the receiver is essentially the same, and details of the receiver printed circuit board and wiring are provided in **Figure 5.** As was the case with the transmitter, check all the wiring very carefully before connecting the unit to the mains supply and switching it on. It is advisable to clearly mark the transmitter and receiver as such using rub-on transfers. Apart from making them look pretty, this will prevent the possibility of the two identical looking units from being accidentally swopped over.

Adjustment

When you are certain that everything is correct, connect both units to the mains

supply and switch them on. Connect the transmitter to the "data out" and ground terminals of the sending computer, and the receiver unit to the "data in" and "ground" of the receiving computer (or whatever). For testing purposes it is probably best to use a simple lopp program to send a continuous series of characters, or even to just repeat the same character indefinitely. Obviously the nature of this program will depend on the computer concerned, but there should be no difficulty in devising a suitable program.

Probably the only adjustment that will be needed is to set VR1 in the transmitter for suitable transmission frequencies, and this is just a matter of finding by empirical means any setting that gives a reliable data transfer. T1 has a very wide bandwidth due to the heavy damping by the output of IC1, and adjustment of its core is unlikely to have a significant effect on performance. T3 also has a fairly broad response, and the receiver circuit is far more sensitive than is absolutely necessary. However, if adjustment of VR1 fails to give reliable results, try using different settings for the core of T3 and the problem should be quickly eliminated. If an audio millivolt meter is available, the cores of T1 and T3 can be given the optiumum settings by monitoring the voltage across the tuned winding of T3, and then adjusting the cores for peak signal level. Of course, when adjusting any piece of mains operated equipment great care should be taken to avoid contact with any mains wiring (which should preferably all be insulated before the unit is connected to the mains supply).

The unit will operate using any word format, but there is a limit to the maximum baud rate that it can handle. The prototype equipment has been tested at rates of up to 1200 baud, but in theory the system should work at up to at least 2400 baud, although it is unlikely to work properly at baud rates much higher than this. The unit does not have provision for handshaking, but when transferring data from one computer to another, or from one data terminal to another, handshaking is normally unnecessary as the receiving equipment should be able to keep up with the data flow. The system should even work properly with a printer provided it has a suitably large buffer, or if a suitable hold-off routine is used at the sending equipment.

PARTS LIST

TRANSMITTER

Resistors (¼W 5%)	
R1	3k9
R2,4	1k
R3	4k7
R5	2k7
R6	47k
Potentiometer	
VR1	10k 0.1W horizontal preset
Capacitors	
C1	1nF carbonate
C2	100nF carbonate
C3	22nF carbonate
C4	10nF carbonate
C5	470uF 10V radial elect
Semiconductors	
Q1,2	BC547
IC1	555
D1,2	IN4002
Miscellaneous	
T1	Toko YHCS11100
T2	Mains primary, 6 – 0 – 6 volt 100mA secondary
FS1	20mm 100mA anti-surge
S1	Rotary mains switch
SK1	3.5mm jack socket

Metal instrument case about 150 x 100 x 50mm; Printed circuit board; Control knob; 20mm chassis mounting fuseholder; Veropins, 6BA fixings, wire, solder, etc.

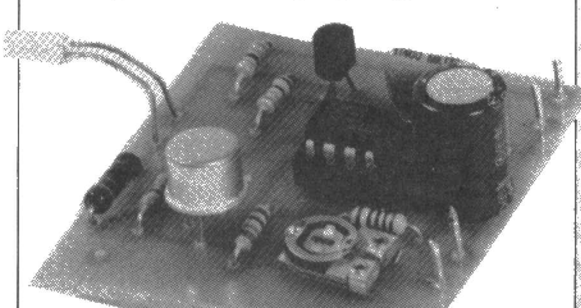
PARTS LIST

RECEIVER

Resistors (¼W 5%)	
R7	22k
R8,9	10k
R10	8k2
R11,12	47k
Capacitors	
??	1nF carbonate
C7,9	2n2 carbonate
C8	4n7 carbonate
C10	330pF polystyrene
C11,14	10nF carbonate
C12	3n3 carbonate
C13,15	470uF 10V radial elect
Semiconductors	
IC2	741C
IC3	NE565
D3,4,5,6	IN4002
Miscellaneous	
T3	YHCS11100 (Toko)
T4	Mains primary, 6 – 0 – 6 volt 100mA secondary
S1	Mains rotary switch
SK1	3.5mm jack socket
FS2	20mm 100mA antisurge

Metal instrument case about 150 x 100 x 50mm; Printed circuit board; Control knob; 20mm chassis mounting fuseholder; 14 pin DIL IC socket; Wire, Veropins, mains lead, 6BA fixings, etc.

WHAT TO LOOK FOR IN OUR JULY ISSUE



ON SALE JUNE 13th

BRITAIN'S BEST SELLING COMPUTER PROJECTS MAGAZINE

Atmos/Oric RS232 interface

An easy-to-build project that adds the universal serial data option to the Oric Atmos. Whether you want to hook the computer up to a Modem or printer, this is the project for you.

View printer drivers

While View is one of the most popular word processor for the BBC micro it has one large drawback. This is the limited provision for sending control commands to the printer. Next month we describe the means by which a large number of special printer commands can be used in conjunction with the View package.

Infra-red data link

Ideal for experimenters in the field of robotics, control systems or for those of you who fancy a IBM PC, like 'wireless' keyboard. The design provides a serial data link at rates of up to 300 baud.

PLUS

Competitions, our regular features, the low-down on all utility software currently available for home micros, and of course, **Your Robot.**

THE INTRICACIES OF INTERFACING

There is no easier way to blow up a computer than by incorrectly connecting up an experimental interface. Paul Beverley, in the first of a new series, has a few hints and tips on the right way to go about interfacing on the BBC micro.

If a microcomputer is to be linked up to the outside world it needs some sort of interface or "port" ie a device through which signals can go in and out. A large number of computers based on the 6502 microprocessor make use of a device known as the 6522 versatile interface adaptor. This is an extremely complex device, probably just as complex as the 6502 itself, and certainly less well documented from the end user's point of view.

Unless you are designing a microprocessor system from scratch, the only thing you will need to know about the 6502 itself is how the various instructions work – and there are plenty of books about that. The 6522 on the other hand cannot simply be dealt with from the software point of view; you do need to get an appreciation of the hardware side of things as well.

Unfortunately, of the many texts that I have seen that try to explain how the chip works, some fail to describe many of the chip's functions sufficiently clearly and even worse, some have a number of both typo-graphical and factual errors. Even the manufacturers' data sheets have some errors and are difficult to understand – At least, I found them so when I first started trying to sort it all out for myself.

'My task is to explain clearly and simply how the 6522 works'

My task then, admittedly not an easy one, is to try to explain clearly and simply how the 6522 works. I shall not try to explain how every bit of it works but only those parts of which I have made a particular study. I have been using the 6522 for almost three years now, and originally wrote about the 6522 in the Atom (who remembers Tim Edwards from the early days of *E&CM*?) and then the BBC micro, and finally I attached one to my Electron. So I have used the VIA quite extensively, but even so I still have not investigated all of its facilities!

An overall view

Before delving into detail, let's get an overall view of the chip and what functions it offers. The summary given in the data sheets is quite succinct:

- Two 8-bit bi-directional I/O ports (bit programmable)
- Two 16-bit programmable counter/timers
- Serial data port
- Peripheral control lines allowing extensive "handshake" facilities
- Latched output and input registers

The chip has four address lines and thus

uses 16 address locations starting from a base address, eg the Atom has a VIA at B800 – B80F, the AIM 65 has one at A000 – A00F, the VIC 20 at 9110 – 911F. The BBC microcomputer uses two of them, one at FE40 – FE4F which is used mainly for internal control (keyboard, sound, speech etc) and the other, at FE60 – FE6F, for linking to external devices. Connection is made to this latter VIA through the Printer Port and the User Port. (Actually, my BBC has three VIA's. I have fitted an extra one onto the 1MHz bus at FCC0 – FCCF. Some people are just never satisfied!).

Associated with those 16 addresses there are approximately 18 registers. The reason why I say approximately is that some registers can be read but not written to directly, some can be written to but not read directly, some registers are written to by more than one address, and some addresses write to more than one register . . . need I continue? It does need a little bit of explanation so let's start with something fairly simple – the input/output ports. This is the facility which is most likely to be used to begin with. For example it could be used to control lights and motors, and to sense the position of an object by using read switches and magnets, or visible or infra red light beams.

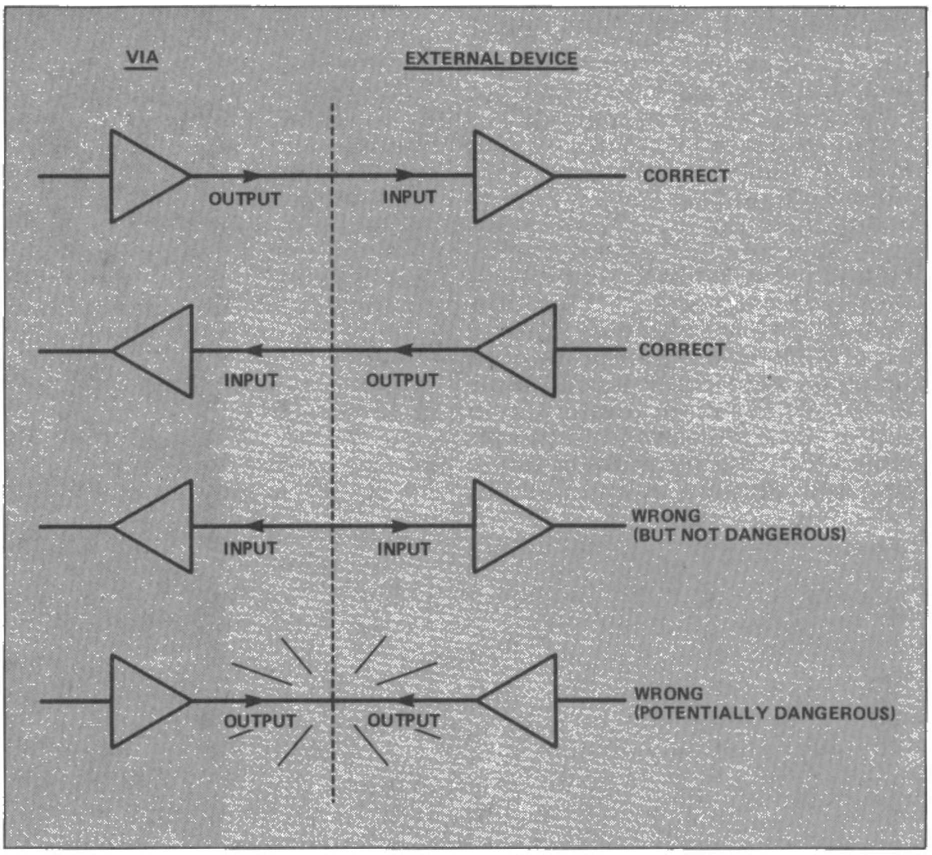


Figure 1. Possible combinations of interconnecting a VIA to an external device.

The in's and out's

There are two 8-bit ports. They are known as PB and PA ie ports B and A. You might have expected A to be first and then B, but the designers in their wisdom named them the other way round ie PB is the first address (&FE60 on the BBC) and PA is the second (&FE61). With apologies to non BBC users I shall continue to refer to the BBC addresses, and will always, unless stated otherwise, be referring to the "external VIA" the User Port (PB) and Printer Port (PA).

To say that these ports are "bit-programmable" means that each individual line of each port can be used either as an input or as an output. This makes for considerable versatility since you are not limited, as with some interface adaptors, to 8 bits of output and 8 bits of input. You could have any combination from the one extreme of having all 16 bits as inputs and vice versa.

However, this programmability means that there could be a real danger of blowing up either your VIA or the device to which it is attached, by getting the programming wrong. However, as long as you think carefully what you are doing, there is no real problem, so let's see wherein lies the danger.

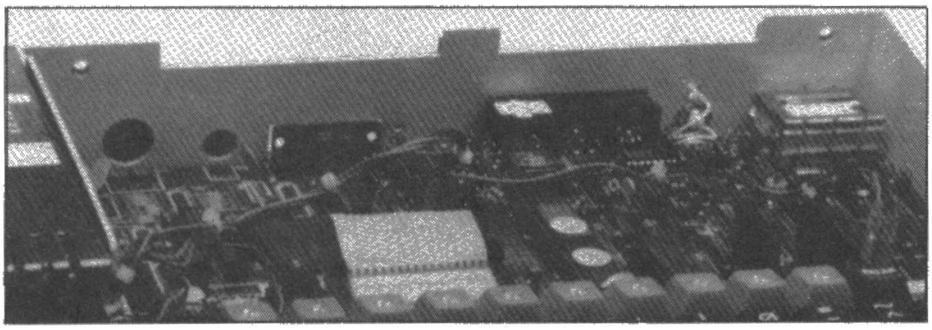
Figure 1 illustrates the situation where you are using your VIA as either an input or an output, connected to an external device which can also act as either an input or an output. The first two situations show what you would want to happen, ie either the VIA acts as an output while the external device receives the logic signal which is sent out, or vice versa.

If however the VIA is set as an input as well as the external device, then nothing much of any interest happens since neither is actually sending signals to the other. Generally speaking you would expect both the VIA and the external device to be registering logic 1 since certainly the VIA and possibly the external device will have a "pull-up" resistor ie a resistor connected between the input line and the positive supply rail so that, in the absence of a circuit capable of drawing current away from the input, it will pull the input up to logic 1. So if you have nothing attached to the VIA and it is acting as input then it will register all 1's if you read it.

'this programmability means there could be a real danger of blowing up your VIA or the device to which it is attached'

It is when the VIA is set to output and we try by mistake to put something into it from the output of the external device that the trouble starts. Each of the outputs is trying to assert a particular voltage level to represent either logic 1 or logic 0. If they both happen to be trying to assert the same level then there's no problem, but if one is trying to pull the line high while the other is trying to make it go low . . . one or other has to lose! Actually it is not always catastrophic, but it is to be avoided if at all possible.

So, as you can see, it is the INPUT state that is the safest one and the output that has to be treated with care. Thus, if you are using for example, three port lines as output and one as input then the four lines that are unused should be set as inputs. I know it feels wrong somehow – by setting the port to input it sounds as if you are "opening up the computer and allowing all sorts of nasty things to get into it". It seems as if we ought to be setting it to output to make sure that nothing can get in, but to repeat this important point, the problem arises if the VIA is trying to put a signal OUT whilst the external device is trying to put a different signal IN. This is why when the VIA receives a reset signal, it sets both ports to the input state.



FEATURE

Getting the direction right

Since it is so important to get the direction right, let's find out how to set the direction before going any further. We will look at port B only to avoid the confusion of trying to talk about both ports at the same time. In terms of programming, Port A is the same as Port B although its electrical characteristics are slightly different (see later).

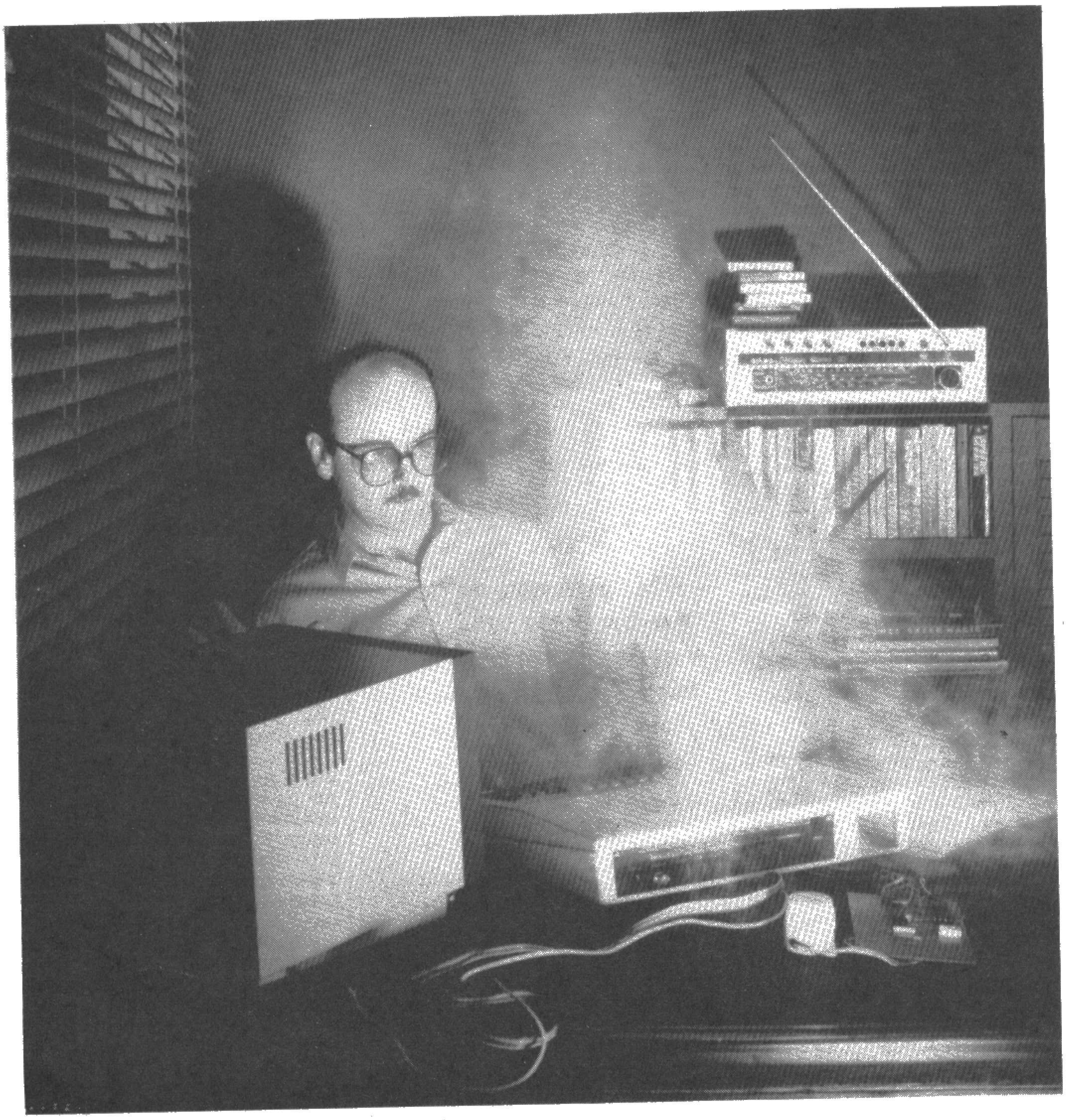
Associated with PB (address FE60) is what is known as the Data Direction Register B (DDRB – address FE62). The eight bits of DDRB relate to the eight bits of the actual port, and each of the bits DDRB0 to DDRB7 acts as a switch to select whether each of PB0 to PB7 acts as either an input or an output. Logic 0 in a DDR bit sets an input and logic 1 sets an output, so if for example DDRB0 is set to logic 0 then PB0 is an input, whilst if DDRB7 is set to logic 1 then PB7 is an output.

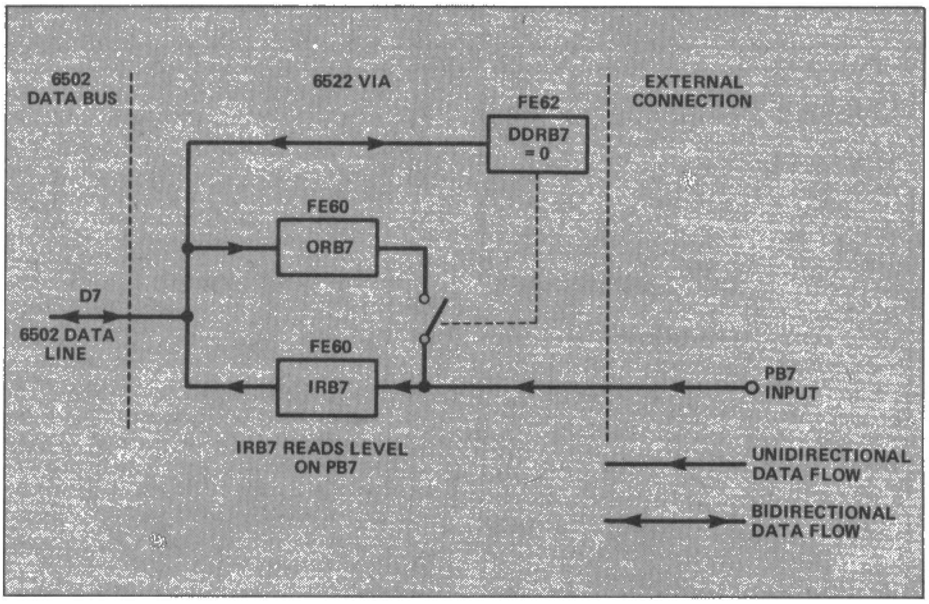
Remembering that having the port lines as outputs is the more dangerous state, what you must do is to decide which lines you actually want to use as outputs, set their corresponding bits in the data direction register to logic 1 and program all the other bits, whether they are actually being used or not, to logic 0. So, for example, if you want to use PB5, PB6 and PB7 as outputs, the PB0 and PB1 as inputs then you would set DDRB = &D0 (1110 0000 in binary) ie *ONLY* PB5 to 7 are outputs, the rest are all inputs.

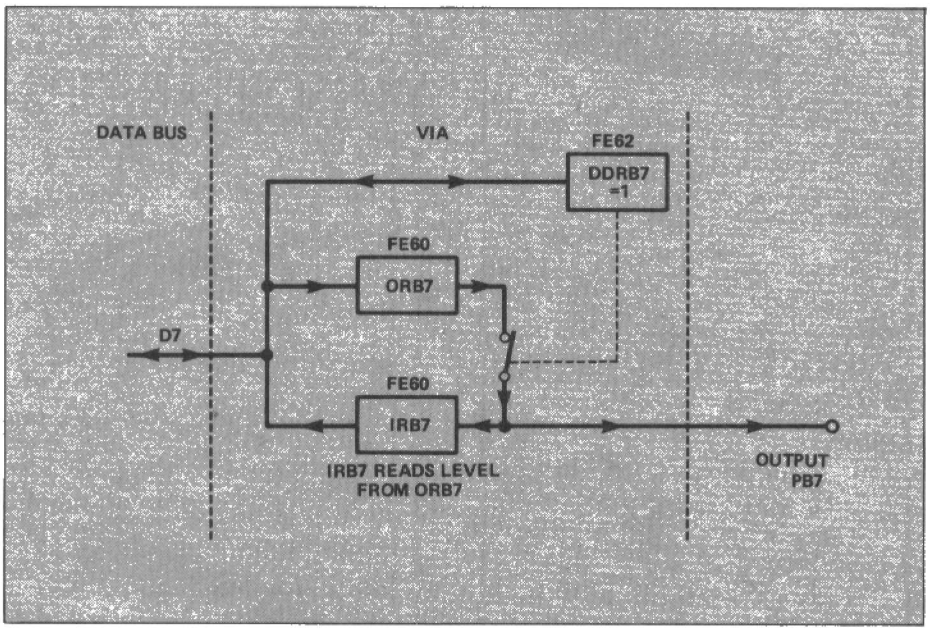
For PA, exactly the same idea can be followed except that Data Direction Register A is at FE63. the only other point to note here is that for BBC micro users, PA is rather special. Acorn, having designated PA as the Printer Port, have added a buffer chip to it in order to increase its current driving capabilities. What this means is that it can only be used for output and *NEVER* for input. The buffer chip will protect the VIA if you do something silly and try to use PA as input, but replacing the buffer chip will cost you almost a couple of pounds, so it is still an expensive mistake to make. What is more, I have noticed that on some pcb's these chips are soldered in and are not in sockets, which makes life a lot more difficult if they do need replacing.

I/O registers

If you look in the data sheet for the 6522,







Figures 2a & b show PB7 set as input and output respectively.

you will find that it speaks of there being two registers at address 0 – the input register B and the output register B. I have seen, in print, the statement that there is no physical distinction between these two registers. This is not true – there is a quite clear distinction. For example, try writing zero into the PB port address.

(?&FE60 = 0 on the BBC micro)

The read the port.

(PRINT ?&FE60)

Assuming that you haven't previously changed the DDRB register and have nothing physically connected to the port then the answer you get will be 255!

Make sure there is nothing connected to the port and then set all eight lines to output by writing 255 into the data direction register. (?&FE62 = 255). Then read the port again (PRINT ?&FE60) and this time the number you put in originally, ie zero, will be returned to you.

Figure 2 represents, in diagramatic form, what is happening. It shows a single bit – bit 7 – of both the port (FE60) and the data direction register (FE62). Each block represents a single bit (bit 7) in the registers. As you can see, the input register bit (IRB7) is permanently connected to the data line so that it can be read at any time, but note that it is ready only ie it is only enabled when the processor reads FE60. The output register bit (ORB7), on the other hand, is a write-only register. Therefore when bit 7 of the data direction register is set to 0 in order to make bit 7 act as an input, (**Figure 2a**) ORB7 cannot be read at all. The value which the processor actually reads is the logic level applied to the external PB7 line. If there is no external level being asserted then the pull-up resistor on the input makes the processor register a logic 1 – hence the 255 (&FF) which was returned when PB was acting entirely as input.

When DDRB7 is set to logic 1, (**Figure 2b**) the output register is connected to the external line and hence also to the input register. Thus in our example, the logic 0 which had been written into the output register and which was un-readable when the port was acting as an input is now capable of being read.

Table 1 – Electrical characteristics of the PB port, PA port and the 74LS244 as used on the BBC micro.

PCB Port of the VIA:
LOGIC 1: Can supply up to 100 μA at 2.4V or up to 1 mA at 1.5V
LOGIC 0: Can draw up to 1.6 mA at 0.4V

PA Port of the VIA:
LOGIC 1: Can supply up to 100 μA at 2.4V
LOGIC 0: Can draw up to 1.6 mA at 0.4V

74LS244 buffer:
LOGIC 1: Can supply 15 mA at 2.4V
LOGIC 0: Can draw up to 25 mA at 0.4V

Electrical Characteristics

As mentioned earlier, there is a difference between PB and PA in terms of their electrical characteristics. PB uses an output stage with two transistors, one to pull the line up to logic 1 and the other to pull it down to logic 0. PA, on the other hand, only has one transistor, for pulling the line down to logic 0 and a resistor is used to pull the external line up to logic 1 when the transistor is switched off. Thus the quoted characteristics for the two ports are the same at logic 0, but PB has a much increased current driving capability at logic 1 compared with PA. (BBC users should remember that their PA port is buffered and that the output characteristics are those of the 74LS244 which is used as an output buffer). **Table 1** gives a list of the electrical characteristics. These figures mean that when designing the external hardware you should not rely on having more than the quoted currents, though in practice more current is often available. As can be seen from the figures quoted for PB at logic 1, as you draw more current, the output voltage drops more, and this must of course be taken into account when designing external circuitry.

I have gone rather slowly through all this but there are many people who would like to get started into the area of interfacing but are unable to start as they do not understand what is going on. I hope that this and future articles will help these people to get started. Those who are in the know will say that I have omitted to talk about "latching" of the input port and the use of control lines, but it is unlikely that beginners will need to know about this and so I am reserving it for a later article. ■

Next Month we will begin to look at the timers and see, amongst other things, how to use them to measure the time of various external events. This could be applied, for example, to measuring the speed of an object cutting through a light beam.

MICROGRAPHIC TECHNIQUES

Mike James draws together the ideas developed over the past few months to develop a program for a three dimensional graphics viewer.

This month's Micrographics is unusual in that it doesn't present any new ideas! Instead a program that uses most of the ideas introduced in previous articles is developed. You could say that this program is the goal of all three-dimensional graphics theory – it is a three-dimensional viewer. That is, given a definition of an object in terms of co-ordinates it allows you to view the object from any position. In terms of practical applications a three-dimensional viewer is at the heart of such things as flight simulators and many computer-aided design packages. For example, in the case of the flight simulator the object being viewed is composed of all the buildings, roads, trees etc. that make up the landscape, and the particular view that is produced on the computer screen corresponds to what would be seen from an aircraft flying over the landscape.

The three-dimensional viewer given in this month's Micrographics is not quite up to the development of a flight simulator – but only because it is written in BASIC and hence not quite fast enough! In fact the speed of the program is much faster than you might expect for such a complex calculation – around a second per twenty display points – and it is almost good enough to give the impression of 'flying' over the object being viewed. The response of the program could certainly be improved by optimising the subroutines that do the calculation but for real speed you cannot beat assembler.

The specification

Before starting to write the three-dimensional viewer it is worth making clear what it should do. (This is a good idea before starting to write ANY program!) The difficulty of defining a three-dimensional object (simply because of the sheer number of co-ordinates that have to be entered) is not a problem that has much to do with the essence of a three-dimensional viewer and so it will be assumed that the co-ordinates that define the object are available in DATA statements. This was the method used in last month's example programs and it would take far too long to improve on it. Given a description of the object in terms of the co-ordinates of its points, the only item of data left to define is the position of the 'observer'. This is not as simple as might be expected. To start with it is not enough to give just the co-ordinates of the observer's position because while this would tell you where the observer was, it doesn't tell you where the observer is looking! This problem of specifying the observer's position and direction of view is discussed in the next section but what is clear is that, even if the co-ordinates of the observer's position were sufficient, this wouldn't be an acceptable way for the user to interact with the program. What is needed is some way in which the user can control the movement of the observer in a 'natural' way without having to worry about numerical co-ordinates. In general when viewing a real object humans tend to think in terms of moving closer or further away, of moving to the right or the left or up and down and these are the three types of movement that the program should try to make available to the user.

"... the polar co-ordinate system is an obvious choice for specifying position ..."

"... you could say that this month's program is the goal of all three-dimensional graphics theory – a three-dimensional viewer ..."

Polar co-ordinates

From the discussion in the previous section about how the user should be able to vary the observer's position it becomes obvious that the usual three x,y,z (or Cartesian) co-ordinates are not appropriate for this purpose. However, if the observer's position is specified using 'polar co-ordinates' then there is an immediate correspondence between the co-ordinates of a point and the distance from the object, moving left and right and up and down. If you are not familiar with polar co-ordinates then **Figure 1** should explain all. In polar co-ordinates the position of a point is specified by giving R, its distance from the origin, TH the angle with the x axis and PH the angle with the x,z plane. (More usual notation than TH and PH are the greek symbols theta and phi). Obviously keeping TH and PH fixed and varying R takes the point closer and further away from the origin without altering its orientation, changing TH while keeping PH and R fixed makes the point 'move around' the origin without altering its distance or its height. Similarly changing PH while keeping TH and R fixed alters only the height of the object but not its distance from the origin.

If the observer's position is specified using R, TH and PH then it is not difficult to see that each co-ordinate could be changed by pressing a key on the keyboard, thus giving the user a natural and interactive control of the observer's position. Although polar co-ordinates solve the problem of specifying the observer's position there is still the small matter of where the observer is looking. If we imagine that the observer is in fact a camera then the question can be more simply put as which way is the camera pointing? The most obvious solution is to make the camera look at the origin because this is the point that the polar co-ordinate system is centred on. The only

trouble is that the object of interest may not be positioned at the origin! The answer is to shift the origin of the polar co-ordinate system to a position that corresponds to a point within the object that we want the camera to look toward at all times – this

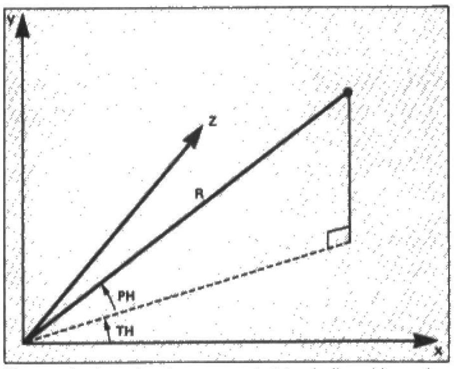


Figure 1 showing how a point is defined in polar co-ordinates.

point is known as the centre of attention (see **Figure 2**).

To summarise –

> *The observer is best thought of as a camera that always looks toward the centre of attention.*
>
> *To specify the view of the object needs both the co-ordinates of the centre of attention CX,CY,CZ and the position of the camera given in polar co-ordinates R,TH,PH.*

Notice that the polar co-ordinates that are used to specify the position of the camera are measured from the centre of attention not the origin. That is R measures the distance of the camera from the centre of

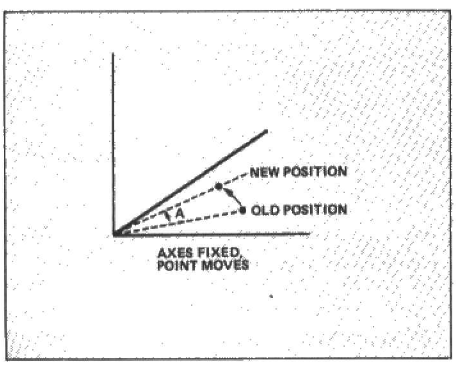


Figure 3a. A rotation matrix can be seen as rotating an object through an angle A.

attention rather than the origin.

View co-ordinate system

Now we have a way of specifying where the observer is and what is being observed the rest is easy – in theory at least! The only remaining problem is that we have a description of the object in terms of co-ordinates that are relative to a fixed set of axes – the object's co-ordinate system – but these are not the co-ordinates that describe the object as viewed from the observer's position. Clearly the observer has a different set of axes that the co-ordinates of the object are measured from – the view co-ordinate system (see **Figure 2**). The co-ordinator of the object in the view co-ordinate system represent the particular point of view that the observer has of the object and clearly they change as the position of the observer and the centre of interest change. To allow the program to show different views of the object

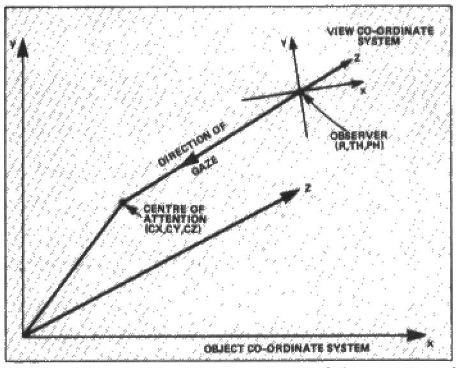


Figure 2 illustrating the concept of the centre of attention.

it is crucial to find a way of converting the unchanging object co-ordinates into the variable view co-ordinates – in fact what we need is a transformation.

The view transformation

Transformations and the use of matrices to represent and implement them have been discussed a number of times in Micrographics but always as a way of changing the position, size and orientation of an object. What we currently need is a way of changing a set of co-ordinates that are relative to one set of axes to corresponding co-ordinates relative to another set. In fact a transformation can be looked on either as a way of changing objects while keeping

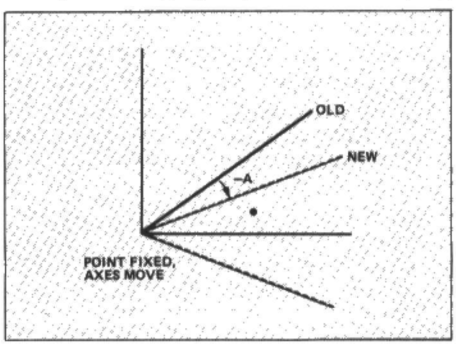


Figure 3b. An alternative view of the effect of a rotation matrix is to imagine the axes rotated through angle A.

the axes fixed or changing the axes while keeping the object fixed – this is a consequence of all motion being relative! For example, if you apply a rotation matrix then you can either think of it as rotating an object through an angle A or you can think of it as expressing the co-ordinates of the object as they would be from a set of new axes obtained by rotating the old axes through an angle –A. (See **Figure 3**).

Now that we know that a transformation can either be thought of as changing the position and orientation of the object or of the co-ordinate axes it is easy to see that what we need for the viewer program is a transformation matrix that will change the axes from the object co-ordinate system to the view co-ordinate system. Such a transformation matrix is known as the view transformation.

Tools for the job

At this point we have all of the ideas that are necessary to implement the three-dimensional viewer. The object or objects will be represented by a point file and a line file as described last month. The observer's position will be fixed by the location of the centre of attention and the polar co-ordinates R,TH and PH. To obtain the objects as seen from the observer's position all we have to do is work out the appropriate view transformation matrix and then apply it to all of the points in the point file. Finally to obtain a two-dimensional representation of the objects we have to apply a projection matrix and then plot the resulting two-dimensional points and lines. Of the two types of projection matrix given last month the perspective projection gives a much more realistic display and so this will be used to implement the viewer. (However, if you are interested, changing the final program to a parallel projection is simply a matter of changing one subroutine). The perspective projection matrix given last month was a projection onto the x,y plane of the current co-ordinate system. This suits our application perfectly as, when the perspective projection is applied to the point file, the current co-ordinate system is the view system and in this the x,y plane is at right angles to the direction in which the observer is looking (see **Figure 2**). In this sense the x,y plane of the view co-ordinate system is like the film in our imaginary camera.

A calculated view

The algorithm for the view program is now perfectly clear. Unfortunately one of its steps – calculating the view transformation given the position of the observer – is remarkably tricky. It's not that the final calculation is problematic, it's just quite difficult to work out what the calculation should be in the first place! Don't worry too much about understanding the details of this section – you can still use the program and develop your own based on it even if you don't follow the derivation of the calculation of the view transformation.

Rather than try to produce a single very complicated matrix for the view transformation it is easier (but still not easy!) to approach the full view transformation as a sequence of similar transformations each one taking us nearer the view co-ordinate

". . the observer is best thought of as a camera always pointing toward the centre of attention . . ."

system. The full explanation of how these transformations are deduced would take far too much space for this article and it is probably not something that anyone can understand unless they have tried to work it out for themselves. However, just to present the final transformation would be something of a cheat after explaining so much of the rest of the program so the steps in the argument will be briefly outlined and if you want to follow what is going on in detail then I would suggest drawing diagrams corresponding to each step for yourself. It is perhaps worth stating that the aim is to find a transformation matrix that will transform the object co-ordinate system's axes into the corresponding axes in the view system.

The first stage in the transformation is a translation to move the origin of the object co-ordinate system. Given **Figure 2** and after a little trigonometry you can obtain the following matrix:

$$\begin{vmatrix} 1 & 0 & 0 & -(CX+R*COS(PH)*COS(TH) \\ 0 & 1 & 0 & -(CY+R*SIN(PH) \\ 0 & 0 & 1 & -(CZ+R*COS(PH)*SIN(TH) \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Following this translation the axes have to be rotated so that the z axis of the object system is aligned with the z axis of the view system. This rotation is itself easier to carry out as two separate rotations. The first is about the x axis and serves to bring the z axis of the object co-ordinate system into the y,z plane of the view co-ordinate system. Once the z axis of the object co-ordinate system is in this position a rotation about the y axis completes the alignment of the z axis. The rotation matrices for these rotations are fairly simple. For a rotation of X degrees about the x axis the transformation matrix is:

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & COS(X) & -SIN(X) & 0 \\ 0 & SIN(X) & COS(X) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

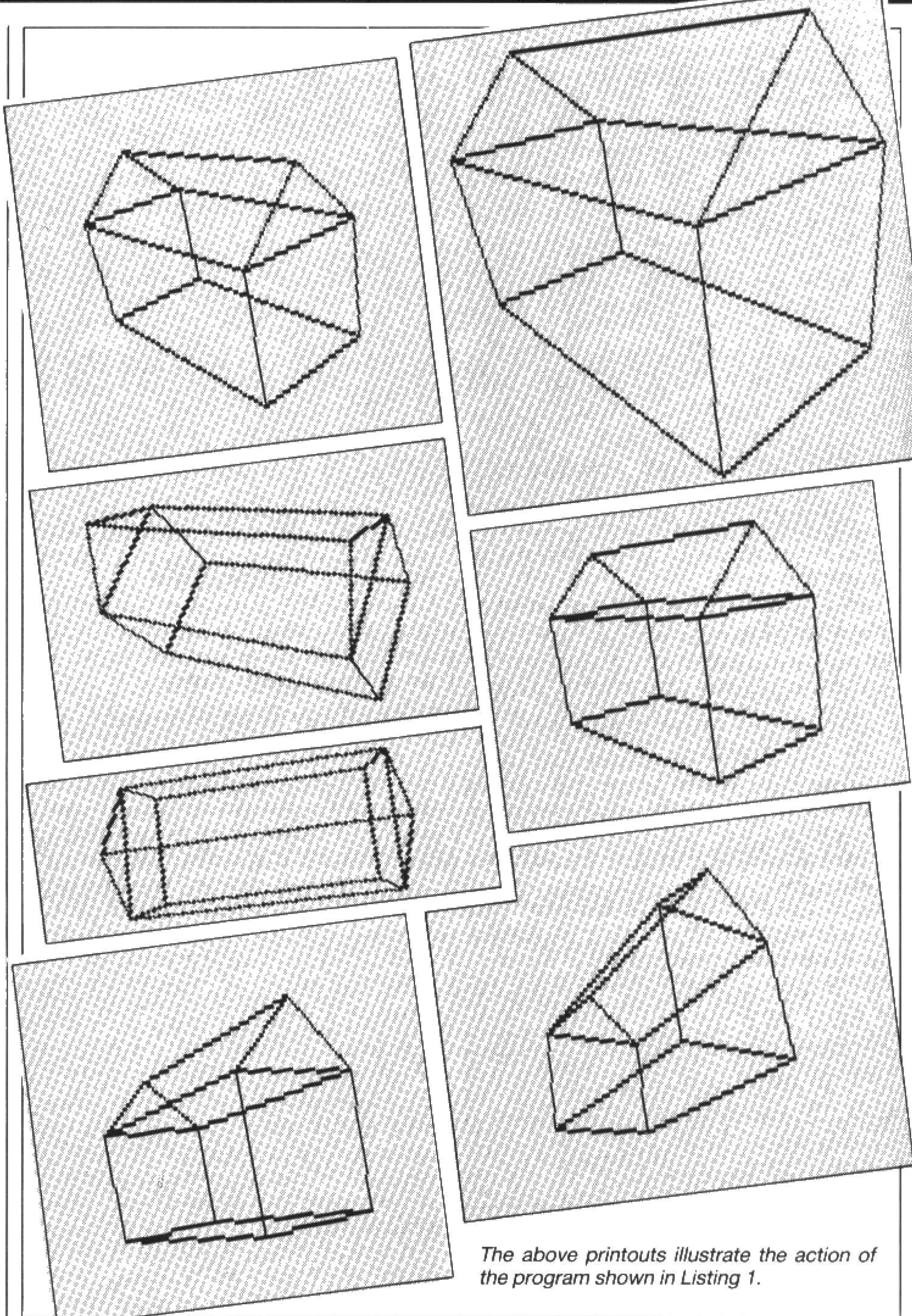
and for a rotation of Y degrees about the y axis the transformation matrix is:

$$\begin{vmatrix} COS(Y) & 0 & SIN(Y) & 0 \\ 0 & 1 & 0 & 0 \\ -SIN(Y) & 0 & COS(Y) & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

The difficulty is working out the angles X and Y! After even more trigonometry we obtain:

```
SIN(X)=-SIN(PH)/SQR(SIN(PH)^2+COS(PH)^2*SIN(TH)^2)
COS(X)=-COS(PH)*SIN(TH)/SQR(SIN(PH)^2+COS(PH)^2*SIN(TH)^2)
and
SIN(Y)=COS(PH)*COS(TH)
COS(Y)=SQR(SIN(PH)^2+COS(PH)^2*SIN(TH)^2)
```

Most of the difficulty in deriving these angles of rotation is due to the choice of a polar co-ordinate system but this is the price that has to be paid to make the program easy to use. Fortunately these trig functions only have to be calculated once for each position of the observer whereas the view transformation is applied to each point in the display file.



The above printouts illustrate the action of the program shown in Listing 1.

After the translation and rotations the origins and the z axis of the two co-ordinate systems are aligned. All that is left to do is a rotation about the (shared) z axis to align the x and y axes. This is simply a matter of applying the following transformation which produces a rotation of Z about the z axis –

$$\begin{vmatrix} COS(Z) & -SIN(Z) & 0 & 0 \\ SIN(Z) & COS(Z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

However once again the problem is to work out what angle of rotation is required. To do this it is worth recalling the idea that the x,y plane of the view co-ordinate system can be thought of as the film in a camera – rotating it through an angle simply alters the orientation of the final photograph on the film. You might think that this doesn't matter too much but if you implement the view program without trying to control the orientation of the x,y axes you will find that the object you are viewing appears upside down as often as the right way up. Clearly it is desirable that the y axis of the view co-ordinate system should be roughly 'upright' ie pointing in the same direction as the object's y axis. The angle of rotation required to ensure that this is the case is even more difficult to work out but fortunately we can let the view program work it out for us by discovering where the object's y axis is pointing after the pair of rotations that align the z axes have been

completed. Once again after much trigonometry we obtain!

COS(Z)=−YA/SQR(XA^2+YA^2)
SIN(Z)=−XA/SQR(XA^2+YA^2)

where XA and YA are the x and y co-ordinates of a point that lies on the object's y axis after being subject to rotations about the x and y axes that have already been derived.

Putting all of these transformations together gives the final view transformation. However there is one last point. To place the centre of attention in roughly the middle of the screen co-ordinate system (recall that this system is the one that your computer uses to plot points) another translation is needed. In practice you need to add about half the screen co-ordinates' maximum x and y values to the final x and y co-ordinates.

The view program

A complete listing of the view program can be seen in **Listing 1.** It should come as something of a pleasant surprise to discover that such a sophisticated three-dimensional viewer turns out to be a fairly short BASIC program. Although written for the BBC Micro-Electron none of the special features of BBC BASIC have been used and it should be possible to convert it to other machines simply by changing subroutine 4000 and the scaling factors used in lines 1035 to 1037 and line 7060. Once you get the program up and running you will see the shape of a simple house. You can view this house from various positions by using the C key to move Closer, the F key to move Further away and the arrow keys to move up and down and left and right around the object. Each time you press D the object is reDrawn from the current position of the observer.

Once you get tired of exploring the house shape you might like to program in your own shapes by changing the DATA statements, lines 1000 to 1224. These record the co-ordinates of P points and L lines in the format described in detail last month but put simply the first P DATA statements are the co-ordinates of the points that make up the object and the next L DATA statements are the start and end point of the lines that make up the object. Thus, line 1100 states that the first line in the object connects point 0 with point 4 (ie the point defined by line 1010 with the point defined by line 1050). If you want to input your own shape then I would suggest that you draw the shape on graph paper before you start typing DATA statements.

You might also like to change the centre of projection defined by lines 1460 to 1480 although the only value worth changing is ZC which alters the degree of perspective. Its current setting at 500 produces a rather exaggerated perspective but this is useful for demonstration. You can also change the centre of attention defined in lines 1430 to 1450. Its current value of 50,50,50 means that the observer is always looking toward a point roughly in the middle of the house.

LISTING 1

```
100 MODE 256
110 WINDOW 512,200,0,0
120 SCALE 1,0,0
130 PAPER 5:CLS
140 mountains
150 clouds
160 bow .5,0,.2,.5
170 tree 1.5,1E-2,.5
180 grass
185 poppies
190 STOP
1000 DEFine PROCedure bow(x,y,r1,r2)
1010 LOCal col,r,s
1020 s=(r1-r2)/7:r=r2
1030 FOR col=2,6,4,5,1,3
1040   INK col:FILL 1
1050   ARC x,y+r TO x-r,y,PI/2
1060   LINE x,y+r TO x,y+r+s
1070   ARC x,y+r+s TO x-r-s,y,PI/2
1071   FILL 1
1072   ARC x+r,y TO x,y+r,PI/2
1074   LINE TO x,y+r+s
1076   ARC x+r+s,y TO x,y+r+s,PI/2
1080   r=r+s
1090 END FOR col
1100 END DEFine
2000 DEFine PROCedure clouds
2010 LOCal i
2020 INK 7
2030 FOR i=0 TO 30
2035   FILL 1
2040   ELLIPSE RND*1.8,.5+.5*RND,RND*3E-2+3E-2,1+RND,0
2050 END FOR i
2060 END DEFine
3000 DEFine PROCedure tree(x,b,t)
3010 LOCal h,w,o,s
3020 h=t-b:w=h/6:o=-w/2:s=-o/2
3030 FOR i=0,16,192+16+2,16+2,2
3035   INK i
3040   FILL 1
3050   LINE x+o,b TO x+o/2,t
3060   o=o+s
3070   LINE TO x+o,b TO x+o/2,t
3080 END FOR i
3090 INK 4
3100 FOR i=0 TO 30
3110   FILL 1
3120   ELLIPSE x+(RND-.5)*h,t+(RND-.5)*h,(RND+.3)*w
3130 END FOR i
3200 END DEFine
4000 DEFine PROCedure grass
4010 LOCal i
4015 FILL 0
4020 FOR i=0 TO 2 STEP 3.5E-3
4030   LINE i,0 TO i,2E-2+RND*5E-2
4040 END FOR i
4050 END DEFine
5000 DEFine PROCedure mountains
5010 LOCal h,x,col
5015 FOR col=3,1
5020   h=RND*.1+col*1.5E-2+3E-2:x=0
5030   FILL 1:INK col
5040   LINE 0,0 TO 0,h
5050   REPeat loop
5060     h=RND*.1+col*1.5E-2+3E-2
5070     x=x+5E-2+RND*.1
5080     LINE TO x,h TO x,0
5090     FILL 1
5100     LINE TO x,h
5110     IF x>2 THEN EXIT loop
5120   END REPeat loop
5125 END FOR col
5130 END DEFine
6000 DEFine PROCedure poppies
6010 LOCal i,h
6030 FOR i=0 TO 2 STEP 5E-3
6040   IF RND>.1 THEN NEXT i
6045   INK 2+4*RND(0 TO 1)
6050   h=RND*3E-2+1.5E-2
6060   FILL 1
6070   ELLIPSE i,h,1E-2
6080 END FOR i
6090 END DEFine
```

The structure of the program can be seen in **Table 1** and you should be able to identify a couple of the subroutines used in last month's demonstration programs. The actual 'number crunching' subroutines are not optimised for speed but are organised to show the steps in the calculation. For example, lines 6010 to 6030 implement the translation, lines 6100 to 6130 implement the rotation about the x axis and lines 6140 to 6200 implement the rotation about the y axis. The final rotation about the z axis that completes the view transformation has to be implemented as a separate subroutine because it needs the result of applying subroutine 6000 to the point with co-ordinates 0,1,0. This point is always included in the point file (by line 1345) as point P−1. Finally it is worth saying that if you have any trouble with the program you should check that you have entered subroutines 5000, 6000 and 7000 correctly as any slight error in the complex calculation will result in rubbish.

Conclusion

This month's program is great fun to use and it has already kept me occupied for many hours constructing three-dimensional objects. It is, however, only the beginnings of a realistic three-dimensional representation of an object. There are the problems of hidden line removal, shading and even stereoscopy to consider. So for next month's Micrographics it is worth trying to find some red and green tinted spectacles of the sort used to view 3-D films etc!

TABLE 1

Line number	Action
10-90	main program
1000-1380	object definition subroutine, lines 1000 to 1310 define the shape of a simple house
1400-1520	data definition CX,CY,CZ is the centre of attention and XC,YC,ZC is the centre of projection
2000-2120	position input subroutine
3000-3040	perspective projection routine
4000-4050	line drawing routine
5000-5110	calculate SIN, COS and other values for the current view transformations matrix
6000-6310	apply the view transformation to the point file
7000-7080	apply the final 'head up' part of the view transformation to the point file and "centre" the centre of attention

See page 57 for details of a special competition linked to the concepts developed during the Micrographics series of articles.

We all know the feeling – having loaded a favourite game you accidentally press BREAK. Or you have a program that you would like to have some some chance of retaining the copyright on. Perhaps Acorn's treatment of the BREAK key as the ultimate ESCAPE was a mistake – but Adam Denning has hacked his way out of it.

BBC breakout!

First, you must forsake BASIC for 6502 machine code, but for most of us that isn't such a problem (!?).

Whenever BREAK (or the even more fatal CONTROL-BREAK combination) is pressed, the BBC Micro corrupts the screen. At first sight it appears that nothing can be done about this, but that isn't quite true if you have some memory to spare. The more useful effect of pressing BREAK stems from the knowledge that it looks for a 6502 JMP instruction at location &287. If it doesn't find one there it carries on with the normal BREAK procedure.

So if we have a JMP instruction in the right location it appears that we can intercept the BREAK key. We can, but there's a little more to it than that. The two bytes following the JMP instruction have to give the address of a routine to handle TWO calls, as pressing BREAK causes this routine to be called twice – once with the carry flag reset and once with it set. The first call occurs before any reinitialisation has occured, and both calls have the interrupts disabled.

To simply protect our software then, all we have to do is put a routine that never finishes in the program, like this:

```
BRKRTN   JMP BRKRTN
```

This simple piece of code will annoy everyone, because the only way to get out of it is to turn the machine off!!

However it appears that we can initialise the three JMP locations by typing:

```
?&287=&4C (this is the JMP instruction)
?&288=BRKRTN MOD 256
?&289=BRKRTN DIV 256
```

I always feel much safer by using the *FX calls dedicated to this purpose. As *FX calls are simply OSBYTE calls, the whole BREAK deactivation procedure can be written into the initialisation routine of your program like this:

This is equivalent to *FX247,&4C,0

```
LDA   #&F7
LDX   #&4C
LDY   #0
JSR   OSBYTE
```

This is equivalent to *FX248,BRKRTN MOD 256,0

```
LDA   #&F8
LDX   #BRKRTN MOD 256
LDY   #0
JSR   OSBYTE
```

and this is equivalent to *FX249,BRKRTN DIV 256,0

```
LDA   #&F9
LDX   #BRKRTN DIV 256
LDY   #0
JSR   OSBYTE
```

and then the rest of your program

Note that all the symbols and calculations are as used by the BASIC assembler, which is not suitable for long programs. See this month's Software Reviews section for a better tool.

Endless loops are not the only form of protection and it would be useful to enable a program to recover fully from a press of BREAK or CTRL-BREAK. This is quite possible but requires a bit of planning as it is very difficult to return to exactly the point that BREAK was pressed. It all depends on the task. For instance, if you were writing a screen editor you could re-enter at the keypress collection stage, if it were a game you would probably need to reconstruct the screen and then use your variables to put the player and any aliens (or whatever) back where they were. You also have to decide on how much machine reinitialisation you want to do. If your program uses any paged ROMs (such as the DFS) then obviously these have to be reinitialised before the program takes over again.

This is easily done. The first call to your routine, with the carry flag reset, is simply ignored. Something like this will suffice:

```
BRKRTN  BCS  BRKNO2
        RTS
BRKNO2  the second call routine
```

This will then cause all paged ROMs to be reinitialised. If a game or utility requires a screen mode other than 7, then there are two ways of doing this. Either use *FX255 to select the Mode upon BREAK, or include this in your second routine:

```
LDA  #22
JSR  OSWRCH
LDA  #your mode
JSR  OSWRCH
```

which is equivalent to VDU22, **your mode** (which is again equivalent to a MODE **your mode** statement in BASIC).

What else should your second BREAK routine do? Well, it is almost certain that your program will need the interrupts re-enabled, so it has to include a CLI instruction, and you may want to issue things like *TV255. Then you can set about restarting your program.

Those of you who read *E&CM* regularly will remember that the April issue contained a Mode 7 editor written in Basic. The machine code version contains this as its BREAK handling routine (in which I decided not to reinitialise any paged ROMs).

```
        BCS  BRKNO2
        LDA  #144              ; *TV255
        LDX  #&FF
        JSR  OSBYTE
        LDA  #22               ; select Mode 7 – this is
        JSR  OSWRCH
        LDA  #7                ; really a 'just in case'!
        JMP  OSWRCH
BRKNO2  CLI                    ; re-enable interrupts
        CLD                    ; just in case!
        LDA  #229              ; set ESCAPE to give ASCII
        LDX  #1
        JSR  OSBYTE
        LDA  #4                ; set cursor keys to give
        LDX  #1                ; ASCII codes
        JSR  OSBYTE
        LDA  #BRKRET MOD 256   ; re-initialise the error
        STA  BRKV              ; handling routine
        LDA  #BRKRET DIV 256
        STA  BRKV+1
        JSR  BREAK1            ; re-initialise the BREAK
        LDA  #12               ; handling routine
        JSR  OSWRCH            ; clear the screen
        JSR  AFTERB            ; not telling!
        JMP  INLOOP            ; return to main program loop
```

Note that most of this routine consists of what might almost be described as in-security, as most of the things I've done do not have to be done each time BREAK is pressed. I did it simply because there is an option within my editor to issue operating system commands, and so this code attempts to remove the affect of any 'daft' commands issued!

Armed with this information, anyone can make their programs virtually BREAK proof. Except that *FX247,0 disables it again! ■

JUNE CHOICE

Disk Systems for the BBC Micro £7.95

by Ian Sinclair

Covers the principles of disk operations, not only for the BBC disk system but also the AMCOM, the Watford and other systems suitable for use with the BBC Micro. Normal and advanced disk operations are discussed in detail and you are shown how to take full advantage of the various disk utilities.

NEWNES BEGINNER'S GUIDES

A popular series of handy information-packed volumes, intended for the comparative newcomer.

- ☐ Bishop Beginner's Guide to Electronics £5.35
- ☐ Fry Beginner's Guide to Computers £5.35
- ☐ Parr Beginner's Guide to Microprocessors £5.35
- ☐ Sinclair Beginner's Guide to Digital Electronics £5.35
- ☐ Sinclair Beginner's Guide to Integrated Circuits £5.35
- ☐ Stephenson Beginner's Guide to Basic Programming £5.35

TEXAS "UNDERSTANDING" BOOKS

A series of informative books designed for "anyone who wants to learn easily and quickly more about today's electronic technology and personal computing".

- ☐ Understanding Computer Science £5.50
- ☐ Understanding Communications Systems £5.50
- ☐ Understanding Digital Electronics £5.50
- ☐ Understanding Microprocessors £5.50
- ☐ Understanding Solid-State Electronics £5.50

Z80 BOOKS

- ☐ Barden Z80 Microcomputer Design Projects £13.95
- ☐ Barden Z80 Microcomputer Handbook £12.85
- ☐ Ciarcia Build Your Own Z80 Computer – Design Guidelines and Application Notes £18.95
- ☐ Coffron Z80 Applications £12.25
- ☐ Khambata Introduction to the Z80 Microcomputer £12.95
- ☐ Leventhal Z80 Assembly Language Programming £15.25
- ☐ Leventhal Z80 Assembly Language Subroutines £15.95
- ☐ Nichols Z80 Microprocessor Advanced Programming and Interfacing £17.95
- ☐ Zaks Programming the Z80 .... £11.25

6502 BOOKS

- ☐ Leventhal 6502 Assembly Language Programming £15.25
- ☐ Leventhal 6502 Assembly Language Subroutines £15.95
- ☐ Scanlon 6502 Software Design £12.50
- ☐ Stephenson 6502 Machine Code for Beginners £7.50
- ☐ Zaks Advanced 6502 Programming £13.45
- ☐ Zaks Programming the 6502 .... £12.95

68000 BOOKS

- ☐ Kane/Leventhal 68000 Assembly Language Programming £16.50
- ☐ King Programming the 68000 .... £9.95

SPECTRUM

- ☐ Angell Advanced Graphics with the Sinclair ZX Spectrum £10.95
- ☐ Angell Advanced Graphics/Spectrum Cassette £9.50
- ☐ Baker Mastering Machine Code on your ZX Spectrum £10.95
- ☐ Bishop Spectrum Interfacing and Projects £7.95
- ☐ Dickens Spectrum Hardware Manual £6.95
- ☐ Hurley Spectrum Workshop – Word-Processing & Beyond £7.95
- ☐ Logan The Spectrum Microdrive Book £6.95
- ☐ Logan The Complete Spectrum ROM Disassembly £10.95
- ☐ Logan Understanding Your Spectrum £7.95
- ☐ Sinclair Introducing Spectrum Machine Code £8.95
- ☐ Street Information Handling for ZX Spectrum £7.95
- ☐ Tang Spectrum Machine Code for the Absolute Beginner £6.95
- ☐ Webb Supercharge Your Spectrum £6.95

BBC MICRO

- ☐ Angell Advanced Graphics with the BBC Microcomputer £10.95
- ☐ Angell Cassettes for Advanced Graphics/BBC £17.50
- ☐ Atherton Structured Programming with BBC Basic £7.50
- ☐ Birmbaum Assembly Language Programming for the BBC Micro £9.95
- ☐ Cryer Basic Programming on the BBC Micro £7.95
- ☐ Cryer Graphics on the BBC Microcomputer £7.95
- ☐ Cryer Cassette for Graphics £12.00
- ☐ Freeman Beyond Basic £8.25
- ☐ Freeman Cassette for Beyond Basic £12.00
- ☐ James The BBC Micro – An Expert Guide £7.95
- ☐ McGregor Advanced Programming Techniques for BBC Micro £9.95
- ☐ McGregor BBC Micro Book – Basic, Sound & Graphics £8.95
- ☐ Smith Interfacing Projects for the BBC Micro £7.95

DRAGON

- ☐ Gifford Making the Most of your Dragon £6.95
- ☐ James Anatomy of the Dragon £7.95
- ☐ James Language of the Dragon: 6809 M/C £7.95
- ☐ Phillips Programming the Dragon for Games & Graphics £6.95
- ☐ Reyden Dragon Machine Language for the Absolute Beginner £7.95
- ☐ Smeed Inside the Dragon £8.95

SINCLAIR ZX81

- ☐ Baker Mastering Machine Code on your ZX81 .... £8.50
- ☐ Costello Advanced Programming for the 16K ZX81 .... £6.95
- ☐ Logan Understanding your ZX81 ROM £6.95
- ☐ Logan Complete Timex TS1000/ZX81 ROM Disassembly £10.95

COMMODORE

- ☐ Gerrard Exploring Adventures on Commodore 64 .... £7.95
- ☐ Gerrard Using the 64 .... £10.95
- ☐ Jones Mastering the Commodore 64 .... £7.95
- ☐ Lawrence Advanced Programming Techniques on Commodore 64 .... £6.95

LANGUAGES

- ☐ Brodie Starting Forth £17.25
- ☐ Brown Pascal from Basic £7.95
- ☐ Christensen Beginning Comal £7.90
- ☐ Ennals Beginning Micro-Prolog £7.50
- ☐ Hogan Discover forth £12.95
- ☐ Oakey Forth for Micros £7.50
- ☐ Ross Logo Programming £9.95
- ☐ Winston Lisp £11.95
- ☐ Wirth Programming in Modula-2 .... £12.95

MISCELLANEOUS

- ☐ Bishop Electronic Science Projects £2.75
- ☐ Browne Disk Drives for Microcomputers £8.95
- ☐ Naylor Build Your Own Expert System £7.95
- ☐ Waite CP/M Bible £17.95

HOW TO ORDER

All prices include post and packing. Just tick the boxes beside the titles you want, fill in the coupon, and send this page to ECM Bookshop. (Or photocopy the page or write out the details). Please note that prices are subject to change without notice.

E&CM Book Service, 45 Yeading Avenue, Rayners Lane, Harrow, Middlesex HA2 9RL
Telephone: 01-868 4854

Please send me the titles ticked.
I enclose cheque/P.O. payable to ECM Bookshop for £ ....
Please charge my Access/Visa account no.

Signed .... Date ....

Name .... (please print)

Address ....

....

....

Hands on the QL

Was it worth the wait? Does it justify the hype? What is that black beetle crawling out of the back of the machine? These were the questions William Owen asked when he finally got to make the pilgrimage to Cambridge for the first 'hands-on' to the Sinclair QL.

The intolerable delay in supply of the QL to Sinclair's customers apparently had two causes. Firstly, according to a company development engineer, the marketing department insisted on launching the machine in January, although they knew it wasn't ready. This was to make a suitable impression on the city prior to the sale of Sinclair shares on the stock market! Secondly, yes, they did have a lot of trouble with the operating system.

Early purchasers of the QL (first batches should be out by now) will notice straightaway a strange, ugly EPROM encapsulated in a crude black plastic package which is stuck into the QL sideways ROM cartridge socket (the beetle). What is it for?

It is an extra 16K of ROM to accommodate the full QDOS. Sinclair reckoned on needing 32K ROM; this was a miscalculation. In fact, a full 48K was necessary. The requisite chips are currently not available; hence the EPROM crawling out of the back of the machine. Sinclairspeak says that the EPROM will be done away with in 2 months (translation: six) and that early buyers will be given an upgrade.

This is not the way to start a review. However, neither are blunders and short cuts the way to market a computer. It is fortunate for Sinclair that their own inimitable cavalier attitude has spawned a computer which is good enough to cover for the mistakes of its creators.

SuperBASIC

SuperBASIC is a clear improvement on that offered by the Spectrum. This is illustrated in Scene a Sinclair written program that shows some of the elegant features of the language, including the way in which any listing maintains the full keywords rather than their minimum abbreviations.

SuperBASIC does have its little problems. It is slower than BBC BASIC – but is also more powerful with a floating point precision of 10 to the power 615 compared to the Beeb's 10 to the 39. Whether anyone will require such precision is, of course, another matter. The BASIC is fast enough for its purposes (much faster than the Spectrum) and speed freaks can naturally make recourse to 68008 codes.

SuperBASIC is very fussy about spaces within program lines – an irritating habit which users will have to familiarise themselves with; yet the error messages are crude and few and far between. All I found were three: BAD LINE, ERROR IN EXPRESSION, and BAD PARAMETER. On pressing BREAK the odd little message NOT COMPLETE appears, and when you do break into a program with the intention of altering a line CLEAR must invariably be typed to reset the stack. This is reasonable, however, because the QL supports both CONTINUE and RETRY after breaking – as of course the BBC does not.

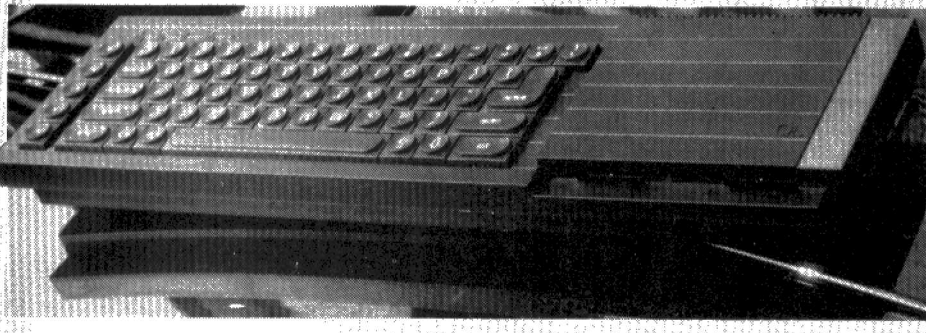
Keyboard

The keyboard is not what it's cracked up to be and don't let anyone tell you otherwise! Programmers will dislike the positioning of SHIFT/CONTROL, to say nothing of the absence of any DELETE key: the latter may however be a blessing in disguise as almost anything can be deleted using the cursor keys and various combinations of SHIFT/CONTROL. If you enter a dodgy line then it will be necessary to retype it, but suitable editors will no doubt soon be rolling off the production line.

The worst factor of all is the key action itself: squashy. Anyone using the word processor will have very tired fingers and wrists after a day's keying in. There is not springback on the keys and the have to be depressed a considerable distance to function – ZX81 to the QL, sublime to the ridiculous.

SuperBASIC

The Microdrives work well. It takes just under a minute to load each of the four QL software packages, and occasionally you have to wait while the drive decides where some of the HELP functions are, but in general the speed is quite satisfactory. Microdrives are in many ways more friendly than full disc drives; there is no kind of 'technofear' associated with them and you can have great fun banging them in and out. LOAD and SAVE operations



worked well and speedily; it is possible to transfer data from one software package to another (if it is entered in the correct EXPORT mode) and to transfer data to the microdrive of another machine over the network.

One big problem with both microdrive and network (and with any other peripheral) is the tedious syntax required to start operation. The key character is an underscore '_', which pops up everywhere. To get a catalogue of all the files on the drive then dir mdv1- must be typed. This isn't too bad but the whole mdv1- must be used in every microdrive access and, for example, you can end up with: exec-w mdv1-'QLtest'prg – horribly irksome.

Visual display

The screen display is excellent. Two windows appear upon powering up. One lists the program, the other runs it. The full screen can be used to run the program by typing CLOSE#2 with a new window statement. Beneath the area under the window the text you have typed appears, and this is structured in the SuperBASIC format in the window above as each line is entered. Users can define their own windows using four simple parameters, with up to twenty windows on each screen. Do not assume, however, that QL windows have any direct relation with the Microsoft or Lisa versions though. On the face of it they are little more than a Sinclair gimmick, but to be fair, programmers may find some very useful applications for them.

Word processing capabilities

The QL is first and foremost a business machine and its software an important component of the system. Hands-on experience proved them to be excellent packages. The wordprocessor and database are superior to other camparable types I have used, though I have no real experience of spreadsheets and business graphics (not that I had time to play with those two).

QL quill (the wordprocessor) is extremely easy to use and at no time did I have to look at the manual to find out how to operate it. It is possible to define your own page format, typeface, etc. and tabulation, margins, underlining etc., require the manipulation of no more than three or four keys. Instructions are present on the screen all the time (they can be removed if you want) with several help function keys to back up the screen display. The only significant problem I found was that in a particular mode the screen update was appallingly slow, but although I could not find how to switch it on or off, there must be a way.

In conclusion

To summarise, I wouldn't cancel my order after seeing the QL, but I'm not sure I'd take any notice of Sinclair the next time they put a new machine on the market. The company have flirted a little too far with the bounds of current technology, and much too far with their customers' patience. The verdict on the QL after a few hours experience is that it looks to be well ahead of the competition, but its reliability remains unproven. ■

Killing bugs

D. R. Bowman has a few hints and tips on how to eliminate the Spectrum's more annoying habits.

The variety and quantity of applications packages available for the Spectrum indicate that it is widely used in situations far removed from its rôle as a games playing machine. In many such applications the colour display of the computer is an unnecessary luxury and, what is more, in many households it would be unreasonable to tie up the family's colour TV set for long periods of time. The solution is often to press a monochrome TV into service as the Spectrum's monitor. In this case however, one of the problems of RF modulated computer signals can become all too apparent.

When I connected a Spectrum to a Ferguson 14" Black and White set, the display, which had been beyond reproach when used with a colour set, left much to be desired.

The whole screen was permeated with interference patterns which might have enthralled a physicist but did nothing for the concentration. On closer examination even the individual characters were seen to be unstable. The results with a Sanyo portable TV gave if anything worse results. Determined to try and find a cure, or at least reduce these problems, I spent considerable time investigating the Spectrum's circuitry. Finally the interference patterns were traced to 30 – 40MHz spurious radiation emanating from the Spectrum breaking into the intermediate frequency amplifier of the TV. The interference was independent of the UHF tuning of the TV and in most cases was present even when the signal/aerial coax cable was disconnected. In fact all leads, including that carrying the mains, were radiating and carrying these signals. I tried filter techniques and results with the Ferguson TV did improve a little. It became obvious that to eliminate the interference, the radio frequency sections of both the Spectrum and its associated TV would have to be bypassed, ie a video output provided on the Spectrum with either a modified TV or video monitor used for the display.

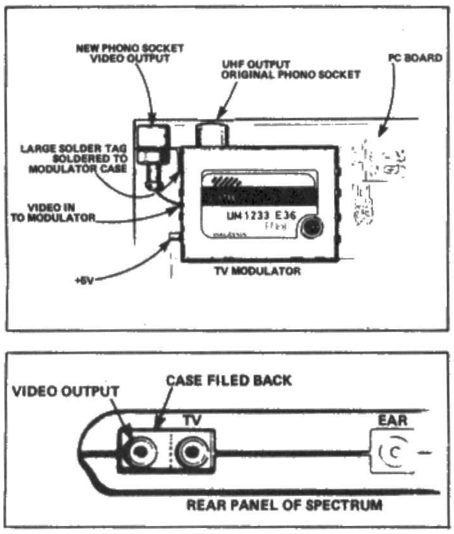
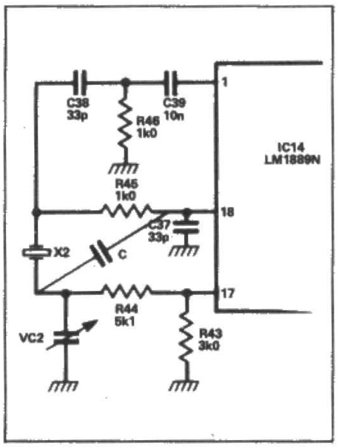


Figure 1. Provision for video O/P on standard Spectrum.



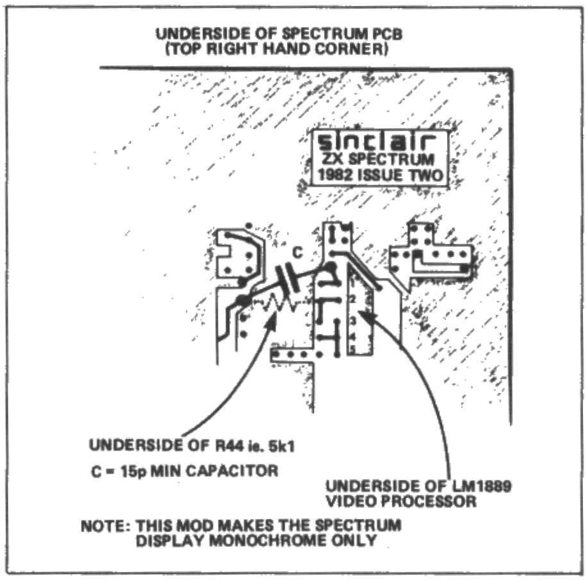


Figure 2a (above). Spectrum B/W display circuit modification.
Figure 2b (right). Modifications to the PCB.

The modification to the Spectrum turned out to be extremely easy. A phono socket was soldered to the left hand end of the ASTEC modulator **(Figure 1),** right next to the UHF TV connector. The connector's outer connection, via a large tag is soldered to the metal case and the inner connection soldered to the video input of the modulator, ie the extra socket is connected in parallel with the modulator's video input and a small section of the black plastic outer case has to be filed away to provide access. This modification can be recommended with only one proviso, ie the VDU to Spectrum connecting lead should be as short as possible, say 1 metre in length, and it is advisable to switch out or disconnect the VDU's input 50 ohm termination resistance.

A few comments are in order vis-a-vis the modification of portable TVs into improvised visual display units. It is most important never to tamper with any TV which does not have a completely isolated power supply and mains transformer.

No video nasties

The change from modulator to video output improved the display considerably. play. All the shimmering interference patterns disappeared, leaving only the shifting character effect visible. Most readers may decide that this result is adequate for their use, but with a small modification to the underside of the Spectrum PCB, even the shifting character effect can be eradicated. This is done by disabling the colour output by soldering a 15p capacitor from pin 18 to the end of R44 (5k1). (See **Figure 2**). This modification does of course limit the Spectrum to monochrome only. It is possible that a switch could be incorporated to allow monochrome/colour change-over facilities.

The computer on which these mods have been made is an example of an early issue two and all or some of these modifications may not be possible on other machines. ▶

E&CM monitor mods

A problem has been experienced with the excellent Spectrum Monitor described on Page 77 of the June, 1983 issue of *Electronics and Computing.* If a M/C program is run and either crashes, or is in the form of a continuous loop, the only way to break-out is to unplug with the resulting loss of all previously keyed-in program material. The Z80 microprocessor has a number of interrupt inputs, among which is the NMI (Non Maskable Interrupt). If this line on the output connector is taken momentarily to ground the computer stops what it is doing almost immediately and goes to address 66H which should be user available. The ROM has an error however and what does happen is pretty useless. If

TABLE 1. The section of ROM which must be altered.

ADDRESS	OP CODE	MNEMONICS
0066	F5	PUSH AF
7	E5	PUSH HL
8	2A B0 5C	LD HL,(5CB0)
B	7C	LD A, H
C	B5	OR L
D	2801	JR Z, 0070
F	E9	JP (HL)
70	E1	POP HL
1	F1	POP AF
2	ED45	RET N

NOTE: USED TO BE AND MUST BE CHANGED TO ABOVE 006D 20 01 JR NZ,0070

TABLE 2. NMI Vector addressing ie 5CB0 50 5CB1 3C

```
10  PRINT USR 15424
20  STOP
```

TABLE 3. M/c Development Memory Area Starts at 7530 H ie 30,000 D

3C40	3E 50	LDA,50
2	32 B0 5C	LD(5CB0),A
5	3E 3C	LDA 3C
7	32 B1 5C	LD(5CB1),A
A	C3 00 3C	JP 3C00

TABLE 4. The Return to Basic called by NMI

3C50	3E C9	LDA, C9
2	32 30 75	LD 7530 ,A
5	C3 70 00	JP 0070

TABLE 5. Basic to enter Modified Monitor Program.

7530	00	NOP
	00	NOP
2	Dev Program	
	C3 30 75	JP 7530
		Test Program End

the NMI routine is changed as shown in the listing **(Table 1)** the MPU will branch to the address to be found at RAM 5CBOH (23728D) and 5CBIH (23729D) least significant bit at 5CBOH. This address can be entered by using an initialising routine consisting of either two POKE instructions or alternatively by an M/C program being added onto the end of the M/C monitor's EPROM **(Ref. 1)** starting at address 3C40 **(Table 2).** The Spectrum will run through this routine and then jump to the original start at 3C00.

To use NMI in this way the area of RAM where M/C development is to take place has to be defined and preferably an address near the start of that area entered into 3C53 and 3C54. In the example **(Table 3)** an addrerss of 7530H ie 30,000D was chosen. If a M/C program crash or infinite loop occurs including this address, then on pressing the NMI button the Spectrum will introduce a RET at 7530 into the program (see **Table 4**) and thus immediately cause a return to basic with the following caption appearing on the screen:

90 STOP statement, 20 : 1

The monitor can be re-entered by the key word RUN followed by ENTER in the normal way assuming that a short program in basic has previously been entered **(Table 5).**

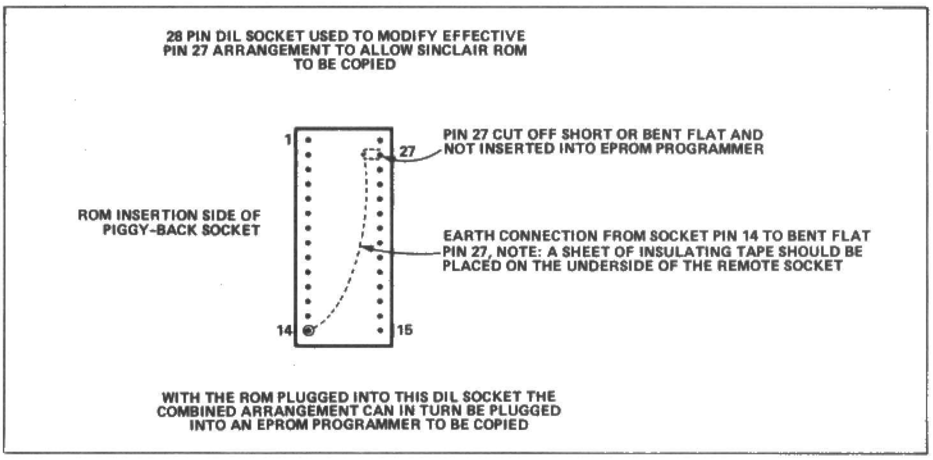


Figure 3. 28-pin DIL socket arrangement.

Spectrum ROM replacement

Mention has been made of one of the errors present in the Spectrum ROM. At first sight this device appears to have a very similar pin layout to that of the 16K EPROM 27128, but further investigation shows that pin 27 on the ROM requires an enabling zero, while the 27128 requires a 1, ie +5 volts. To read the ROM into an EPROM programmer it is only necessary to mount the device in a 28 pin socket (see **Figure 3**) where pin 27 has been cut short and carefully linked to pin 14. It is essential when mating the adaptor to the programmer that pin 27 does not make contact with the programmer's pin 27, but instead is connected via pin 14 to 0 volts. Once this has been done it is a simple matter to modify the ROM's software and to program the new 27128 device. This EPROM will not as it stands function if plugged into the Spectrum and an extra inverter has to be introduced between MREQ and pin 27 of the new ROM.

The add-on is constructed of a small piece of double sided printed circuit board using a 28 pin header, 28 pin socket and one 14 pin socket. Note that all pins of the larger socket penetrate through the board with the exception of pin 27 which is bent flat outwards and soldered to the top of the board. The header connections are all soldered to the protruding socket pins with the exception once again of pin 27, which is only soldered to the under track on the board. This arrangement allows a 7400 inverter to be introduced between the Spectrum ROM socket and the 27128 EPROM. When complete the arrangement can be plugged in by mating the header pins with the ROM socket.

The ability to program an EPROM and use it as a replacement from the original ROM allows us to correct the error noted in Table 1 and also carry out the program

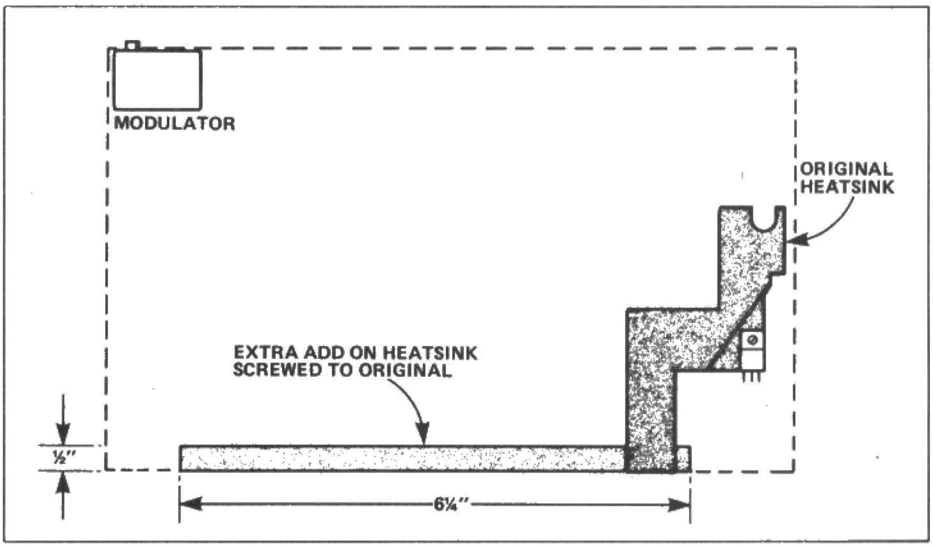


Figure 4. Details of extra heat sink modification.

change outlined in Dr. Ian Logan's Books **(Ref. 2)**.

There are a number of ways of mounting this extra integrated circuit.

a) A simple add-on board is plugged into the original ROM socket with the accompanying disadvantage of having to raise the Spectrum's keyboard.
b) Modifying the main computer board and introducing a small flying lead board rather in the manner of Sinclair's own "Dead Cockroach" modification.
c) Possibly building the board externally and connecting it to the O/P edge connector.

The first of these procedures is to be described together with a practical arrangement for raising the keyboard to make enough headroom for the "piggy back board". If the reader feels a reluctance to proceed thus then either the second or third suggestion can be tried.

As already stated there is not enough headroom in the standard Spectrum for the EPROM board. The best method of overcoming this is to raise the keyboard about 13mm by using spacers and replacing the 5 self tapping screws with 1" long 6 BA screws. This arrangement works well but leaves a gaping slot between the two halves of the Spectrum. It is quite feasible to cut a length of 20 SWG aluminium sheet 13mm wide and with skilled bending make a belt which will fit into the groove round the Spectrum's joint. Take care not to allow the aluminium to short circuit any internal connections. It is convenient to mount the NMI push button and Mono/Colour switch on this aluminium side strip; this avoids the necessity of having to drill hideous holes in the Spectrum case.

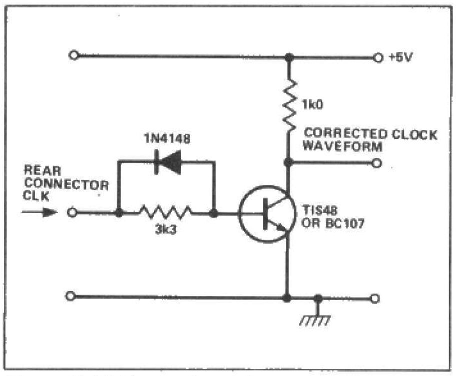


Figure 5. CLK waveform shaper.

A bigger heat sink

On examining the Spectrum's inner workings it comes as quite a shock to find how hot and very small the power supply heat sink is. It is not difficult to nearly double the surface area of this sink by bolting a thin strip of aluminium sheet on to the present sink and running it along the front edge of the case **(Figure 4)**. Care must be taken not to short circuit any of the electronic components and in particular the row of decoupling capacitors connected along the row of 4116 memory chips.

Illustrations of the Spectrum's rear connector appearing in the operator's manual and elsewhere have two mistakes. The first is the so called −12 volts which is in fact an unsmoothed +12 volt line and the second is the CLK output which is a base drive and not a normal clock waveform. This is necessary to generate the very fast risetime required by the 4MHz Z80 CPU. The circuit **Figure 5** should be used whenever an external clock is required for say a Z80 PIO (Parallel Input/Output Device).

This concludes the description of the Spectrum modifications, but a warning is in order. Any tampering with the Spectrum outlines in this article immediately invalidates Sinclair's guarantee and must only be carried out by someone with considerable previous experience.

References

1. Spectrum Monitor, June 1983. *Electronics and Computing.*
2. The Complete Spectrum ROM Disassembly by Dr. I. Logan and Dr. Frank O'Hara.
   Understanding your Spectrum by Dr. I. Logan.

FEATURE

MONITORS

Everything you ever wanted to know about monitors buyers guide and technical explanation of their inner workings by Liz Gregory.

No matter how sophisticated a computer's colour graphics circuitry may be, the quality of the display it ultimately produces will, in most cases, depend on the monitor used as the display terminal. Given the increased emphasis on a home computer's 'pixel power' it is not surprising that many people are beginning to look at their present monitor with a critical eye. Gone are the days when the chunky graphics of a ZX80 displayed on a black and white TV set would satisfy even the most humble of computer users.

Almost all of the popular micros make provision for a dedicated monitor to be used in conjuction with them, although some are more flexible than others in the types of outputs available.

Class structure

In general monitors may be divided into one of three distinct classes depending on the form that the video signal linking monitor and computer takes. The types of monitor are RGB, capable of providing the highest resolution, composite video, and standard TV sets tuned to the frequency of a computer's RF modulator – this system produces the poorest results. In some cases a monitor may offer more than one type of input and indeed some form of direct video input is becoming a popular option on many mass produced TV sets.

Before discussing the ways in which the classes of monitor differ, it would be helpful to outline the way in which images are produced on the screen.

'Almost all micros make provision for a dedicated monitor'

All monitors adopt a system known as raster scanning (see **Figure 1**). In this system an image is built up from a modulated electron beam moving in a series of horizontal lines that traverse the screen from a starting point in the top left hand corner. The fact that an image is formed from 625 lines will not come as too much of surprise but, as ever things are not as simple as they seem. In practice the 625 lines of a full image are made up from two scans of 312½ lines. There are two inter related reasons for this. The first is that in order to produce a flicker free display it is necessary to build up complete scans at a rate of about 50 times a second (this is linked to the behaviour of the human eye). In order to meet this requirement in broadcast terms would mean an excessive bandwidth requirement. By sending two fields of 312½ lines the bandwidth requirement is halved.

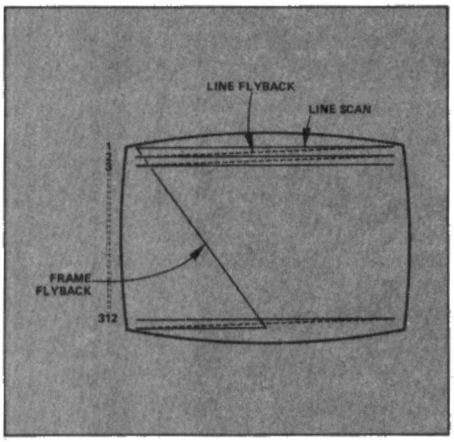


Figure 1. The image on the screen of a monitor is built up from a series of horizontal line scans. At the end of each line the electron beam 'flys back' to the start of the next.

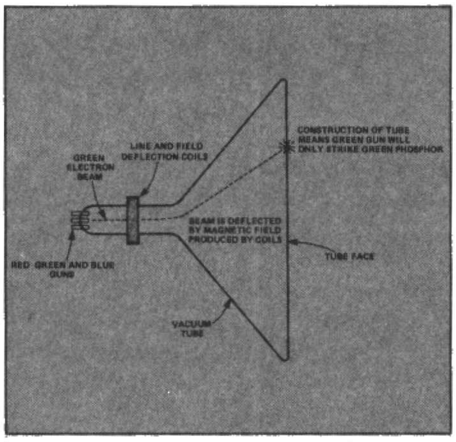


Figure 3. The electron beam emmited from a monitor tube's guns is deflected in both the horizontal and vertical planes by a set of coils mounted round its neck.

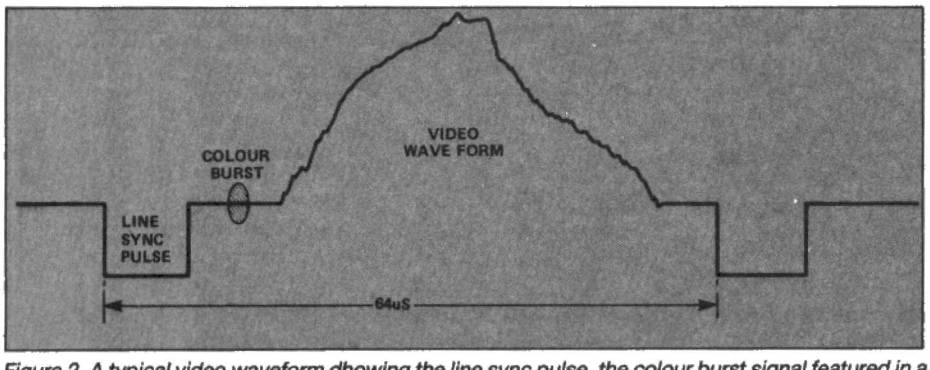


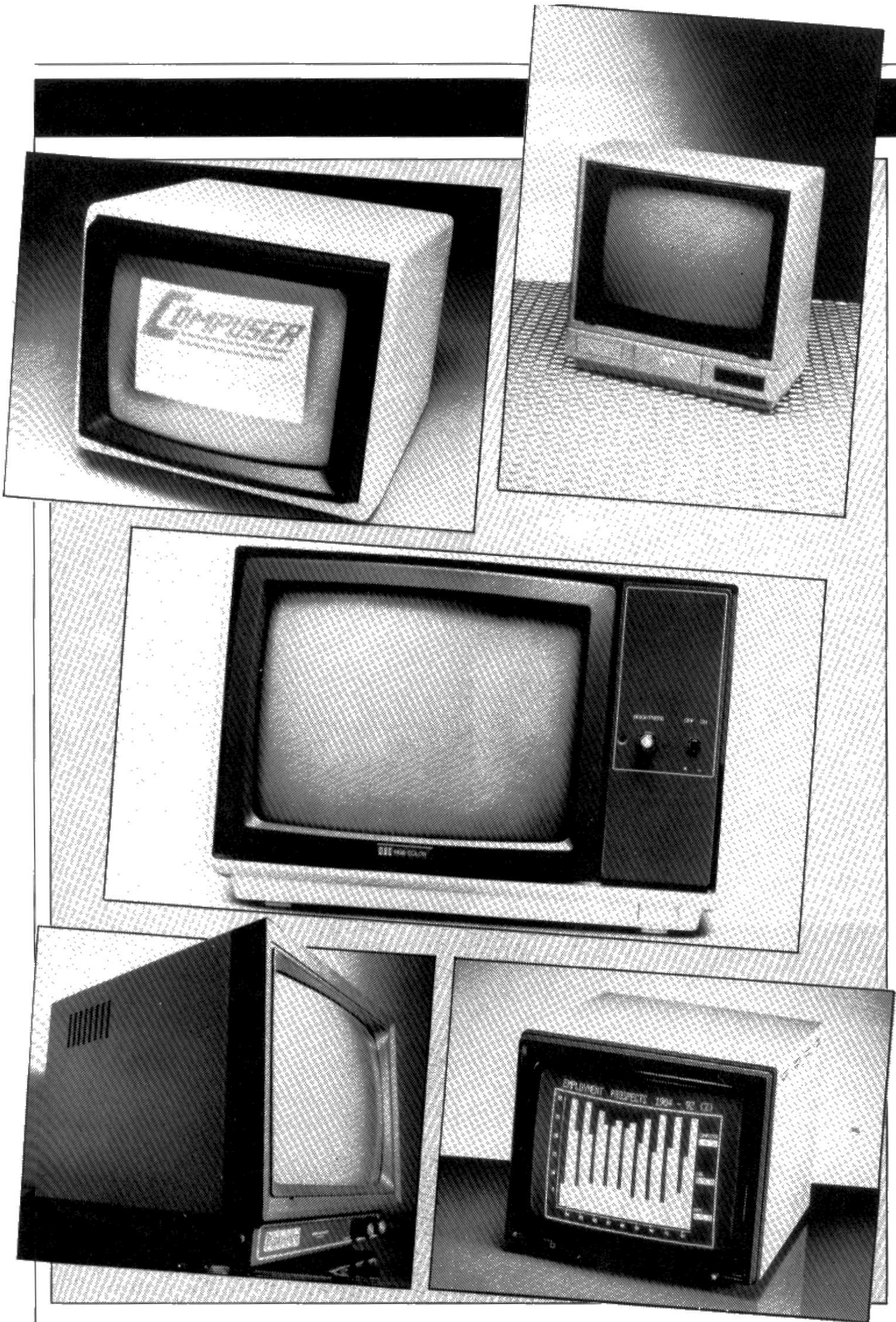
Figure 2. A typical video waveform dhowing the line sync pulse, the colour burst signal featured in a composite signal and the video waveform that conveys the luminance information.

In order to produce an intelligible display it is necessary that the scanning electron beam is synchronised to the computer's video circuitry. In order to achieve this a series of sync signals must be added to the raw video waveform. Two types of sync pulse are required: line sync to tell the monitor when to initiate a new line scan – these occur every 64uS (20mS/312.5=64uS); and a pulse to initiate a new field scan – these occur every 20mS (1sec/50 scans=20mS). (See **Figure 2**).

The scanning principles outlined above are common to all monitors. The differences occur in the way in which the video and sync information is passed to the computer. All colour TV displays make use of the fact that a complete spectrum of colours can be synthesised by combining varying quantities of the primary colours red, green and blue.

In simple terms a computer producing a colour display will store information about the red, green and blue content of a pixel in an area of memory that is separate from, but overlaying the memory used to store the brightness, luminance data. An RGB monitor is capable of accepting this colour data directly; the three guns of the monitor's tube (one gun for each primary colour) are individually accessible. (See **Figure 3**).

In a composite video monitor the information regarding the colour content of the signal is superimposed on the video signal using some rather cunning electronics that allow a single wire to pass all of the required information. A drawback of this system is that a number of compromises



Monitors are currently available from, clockwise top left, Compuser, Fidelity, Digivision and Microvitec.

have to be made and the quality – although better than average – is not quite as good as that offered by an RGB monitor.

The last class of monitor is simply a domestic TV set that has had its tuner adjusted to the frequency of a computer's monitor (usually channel 36). This system, although the most common, offers the poorest quality by virtue of the fact that the signal must be modulated when initially received and then demodulated within the TV, with each signal processing operation degrading the quality of the signal.

The results produced by the varying types are to some extent determined by the way in which the signal is connected to the computer. The quality of the components used in a monitor's construction is also of paramount importance. A measure of a monitor's quality is the figure quoted for its bandwidth or alternatively, the maximum horizontal and vertical resolution quoted in pixels. This latter specification is the more relevant in computing terms as it is far easier to relate to the performance claimed for the computer to be used with the monitor.

Models available

The advantages of a specialised monitor have become more apparent recently. Initially flickering screens made TV sets a better medium even though the rest of the family were inconvenienced! However, as more companies cater for the needs of the personal computer user, prices are coming down and quality improving. RGB failities are now common to most colour monitors, whereas lower priced composite videos were once favoured. Models like Fidelity's CM14 at £199 are the exceptions to the rule of plus £200 pricing, and many people may feel that it is a lot to pay for a specialised monitor in comparison with a TV set.

In this case, TV manufacturers can capture quite a substantial slice of the computer peripheral market. Fidelity, Phillips and Sony already produce cheap, colour TV sets which will suffice for both addict and enthusiast alike and there are signs that companies are designing their sets specifically with this market in mind. For example, Sony have produced their KV1430 Trinitron colour TV with an RF socket at the front of the set so that there is no need to fiddle around the back. They also have raised the screen a few inches off the table level so that, when in use, the computer will not obscure any of the screen. At £249.00 it seems a very reasonable idea.

In order to keep prices down, companies can often adapt TV sets and convert them to monitors. DoubleMode offer a colour monitor for around £100.00 and for a few pounds more provide a model with RGB, composite video and green screen switch. All this for the price of a good monochrome monitor. However the drawback with converted TVs is that the signal from a computer may not adapt as well as with a monitor, therefore the picture may be slightly affected.

Monochrome or green/amber screen monitors are still popular with the micro user, as with educational establishments and word processor users. Most models come ready equipped with anti-glare filters and are generally high-resolution. Digivision's 12" PM monitor is priced quite high (at £112.00) in comparison with the Chiltern Electronics device with a green screen high resolution HM123 costing only £90.00. Most monochrome monitors fall within the £90-£120 price range although Display Distribution's Novex is around £77.00.

Colour monitors have also fallen in price, although as usual you get what you pay for. Despite this some models appear to offer good value for money: Opus Supplies' JVC model features a resolution of 370 x 470 pixels for around £180.00 and this is compatible with most computers. A slightly dearer model affords a greatly increased 580 x 470 resolution, but it ought to be remembered that any graphics created have to be within the realms of the computer's limitations. The Novex-1414-CL comes at just under £200.00, a design which was originally for the BBC but which is now compatible with any computer producing RGB/composite video signals. Again an added bonus is the green screen switch which allows for the use of word processing packages. Slightly more expensive models like the Kaga set of monitors, distributed by Data Efficiency offer the bonus of 16 colours or more in their specification, which is again at the price of a cheaper colour TV.

As interest in home computer and education has grown, so companies who have previously looked to the high technology and business side of computer are changing, moving their outlook and bringing their quality and expertise into the home com-

Maker/Supplier	Model	Price	Colour/Mono	Video Form	Size	Res
Kaga-Data Efficiency	K12G	£109.00	Mono-Green	—	12"	High
	K12A	£119.00	Mono-Amber	—	12"	High
	K12R1X	£215.00	Col	RGB	12"	Med
	K12R2	£285.00	Col	RGB	12"	High
	K12R3	£399.00	Col	RGB	12"	Super Hi-Res
Digivision	PM9/1	£124.19	Mon	—	9"	High
	PM	£112.12	Mon	—	12"	High
	PC14	£241.00	Col	—	12"	Med
Microvitec	1431/MS4	£199.00	Col	TTL	14"	Med
	1431/MS3	£225.00	Col	TTL (for Spectrum)	14"	Med
	1451/MS4	£299.00	Col	TTL	14"	Med
	1451/MQ3	£299.00	Col	TTL (for QL)	14"	Med
	2031/CS5	£299.00	Col	TTL	20"	Med
Grundig	C74000	£410.00	Col	—	14"	High
Novex (Display Distribution)	NovexD/S	£199.95	Col	RGB/PAL	14"	Med
	NC-1418-RH	N/A	Col	RGB	14"	High
	N12/800	£76.62	Mono	—	14"	High
Double Mode	DRC	£97.75	Col	RGB	20"	Med
Compuser	—	£265.00	Col	RGB/TTL	—	Med
Amdek	Colour 1V	N/A	Col	RGB	13"	High
JVC Opus Supplies	JVC	£179.95	Col	RGB	14"	High
	JVC	£229.00	Col	RGB	14"	Med
Cabel	CE370A	£229.00	Col	RGB	14"	Med
Crofton	PM102	—	Mono-Green	—	12"	High

puter field. Compuser anticipate that users want to use their computers for more serious purposes than games, and offer – at around £265.00 – a monitor which has a separate composite video signal for computers with both RGB and composite video output. Digivision are especially concerned with education and offer quality monitors like their PM14/1 at discount prices to educational establishments.

Boost for Spectrum owners

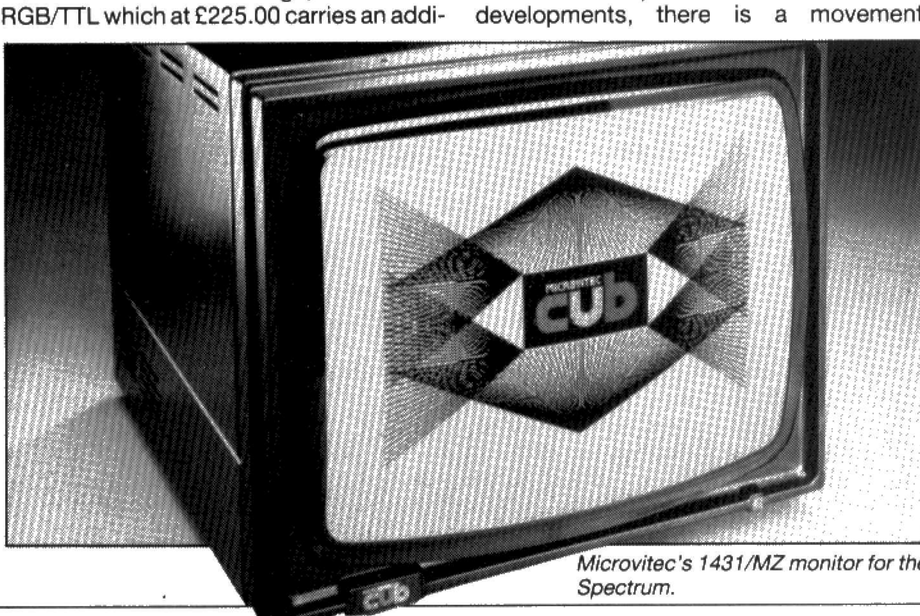
One of the major worries for Spectrum owners wishing to purchase a monitor is the need to buy an interface unit as well as a monitor. Microvitec now offer, alongside the rest of their Cub range, a 1431/MZ RGB/TTL which at £225.00 carries an additional card which "converts" the RGB/TTL output for the Spectrum as well as being compatible with other computers. The company are also aiming their 1431/MQ3 model with 653 x 585 pixels and 18Hz bandwidth at the lucrative QL market.

Update

The increasing interest in dedicated monitors, particularly in the educational field, has naturally decreased prices as competition grows. Fortunately, this can only mean good news for users as quality products are now available at lower prices with most colour monitors featuring RGB as well as composite video. As for future developments, there is a movement towards liquid crystal displays which, although more expensive, are lighter than the ordinary cathode tubes common to TV sets. RH Electronics have brought out a digitiser for the BBC which requires a video recorder or camera for operation and there are more of these on the way.

In addition it is obvious that monitors are regarded as an integral part of the microcomputer: an attitude that has been more common to business computer manufacturers in the past. Witness the new Amstrad reviewed in last month's *E&CM* which offers a monochrome monitor within the £200 price, or a colour monitor for an additional £100.00.

The interest of TV manufacturers who can now provide a direct input to over-ride the RF tuner, the decrease in prices and the influx of monitors for the home market from the bigger manufacturers are all factors which combine to make monitors extremely viable peripherals.



Microvitec's 1431/MZ monitor for the Spectrum.

USEFUL ADDRESSES

DoubleMode
16 Macadam Place
South Newmoor Industrial Estate
Irvine
Ayrshire
Scotland
KA11 4HP

Opus Supplies
158 Camberwell Road
London
SE5 0EE

Microvitec
Futures Way
Bolling Road
Bradford

Data Efficiency
Finway
Hemel Hempstead
Herts

Display Distribution
35 Grosvenor Road
Twickenham
Middlesex
TW1 4AD

Digivision
163 Parker Drive
Leicester
LE4 0JP

Micro Peripherals
69 The Street
Basing
Basingstoke
Hants
RG24 0BY

Compuser
27 Vulcan Way
New Addington
Croydon
CR9 BOBJ

Chiltern Electronics
High Street
Chalfont St. Giles
Bucks
HP8 4QH

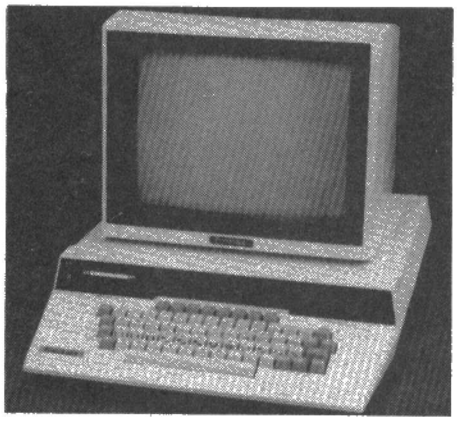
TATUNG'S EINSTEIN

Have Tatung identified a gap in the micro computer market? Gary Evans has the answer.

If Tatung's Einstein computer is to break into the mass micro computer market, it must both steal sales from products already established in the arena and attract customers who, to date, have only thought about acquiring a machine but have not taken the plunge. To this end the machine has been designed to meet the needs of vertical sector of the market which, according to the Tatung management, is not catered for by existing designs. The company see the gap in the market as lying between the 'home' micros in the £250-£500 price range and the 'business' machines that, even in their cheaper incarnations, tend to have price tags in excess of £1,500. To this end the Einstein has been designed to offer a 'computer package', complete with floppy disk drive, at an entry point price of £499. Add a colour monitor and an extra couple of drives and the price is still below £1,000.

Like the new Amstrad CPC464 computer, the design team of the Einstein have adopted a safe rather than technically innovative approach. This similar design philosophy does not come as too much of a surprise as both Amstrad and Tatung have entered the computer market from TV/audio industry where the emphasis is on value for money and reliability rather than on the technology used within a product.

The principle semiconductors of the Einstein would tend to suggest that the designers had the MSX hardware specification in front of them during the development of the machine. The Z80A, running flat out at 4MHz, was the natural choice for the CPU, for all the usual reasons of software availability and 'field experience'. The video display processor is the TI TMS9129 while sound generation is the responsibility of GI's AY-3-8910. Other ICs worthy of note are the Z80 PIO and CTC, the ADC0844 A/D converter and WD1770 disc controller and finally the 8251A which takes care of the RS-232C serial port.



CPU and memory

There is little to say about the choice of the Z80 as processor apart from repeating the above comments about the wealth of programming experience on this MPU. The memory incorporated within the Einstein's design acknowledges the fact that for a computer to be attractive to purchasers in today's market it is almost essential that 64K of RAM be part of the basic specification. The Einstein in fact goes a little further in that a separate 16K of RAM is dedicated to the video display. Overlay techniques ensure that the full 64K of user RAM is available for user software. Further switching within the machine allows the internal 8K ROM operating system to be expanded to 32K.

On display

The choice of a 40 column x 24 row display as standard continues the safe theme of the design. While this will be a limitation in certain circumstances, the 40 x 24 format is the most sensible choice for machines to be used with domestic TV sets and is compatible with both Teletext and Prestel/Micronet. An alternative text mode allows a 32 x 24 in 16 colours.

By virtue of the fact that the Texas video processor was chosen for the Einstein the machine is able to provide 33 individual display planes on which up to 32 sprites may be defined. This makes the generation of complex graphics displays a painless task.

Ultimate resolution of the graphics display is 256 x 192 pixels defined as one of 16 colours.

'the design team of the Einstein have adopted a safe rather than innovative approach'

Disk storage

It is the inclusion of a 3" compact floppy disk drive that sets the Einstein apart from other computers in the £500 class. The drive chosen is the Teac model which offers a capacity of 250K bytes per side unformatted. The disks can be flipped over meaning that a single disk can store 500K.

The DOS and interface of the Einstein will also support standard 5¼" drives.

The BASIC and DOS

Einstein's DOS was developed by Crystal Research and was conceived with the first time user in mind. The commands are clear and easy to understand and the error messages are meaningful phrases rather than obscure codes. The DOS has the ability to run CP/M programs, and indeed a number of CPM packages have already been cross ported to the Einstein's 3" disk system.

Crystal Research were also responsible for the BASIC which is an enhanced version of the company's Xtal BASIC 3. It provides over 190 commands and functions, many of which are used to control the video and sound circuitry.

The BASIC has been designed in such a way that the reserved word and error lists may easily be extended by the user in order that the language can be tailored to individual needs.

Input and output

The Tatung computer has a number of I/O ports including both RS232 and Centronics capability. Another welcome feature is the inclusion of a fast 4 channel A/D converter with an eight bit resolution. Expansion is catered for by a 60 way connector designated the 'Tatung Pipe'. This is in fact simply a means of accessing the Z80's data, address and control lines via a series of TTL buffers.

Provision is made for the addition of two external disk drives which, as mentioned above may be of the 3" or 5¼" variety.

As far as video output is concerned, the Einstein provides a standard RF modulated output in addition to the less familiar YUV format. Using the YUV signal in conjunction with Tatung's £240 TM-01 monitor gives both a high resolution display together with the means of user adjustment of colour saturation. By means

of internal links the YUV output can be reconfigured as an RGB signal.

Sound

The sound generator provides three tone and one noise channel which may be defined in terms of pitch, amplitude and duration. The internal speaker of the computer is larger than found on most micros (3¼" x 2¼") and with a 0.25W amplifier driving it the sound quality is far better than one has come to expect from a micro computer.

Power supply

The Einstein makes use of Tatung's experience with switch mode power supplies to build in a PSU that is both small and cool in operation while providing a high degree of mains filtering.

In Conclusion

The Einstein is a well made computer and the company are stressing that, by virtue of the fact that the machine was designed and is to be built in England, there will be no lack of support for the product. Tatung have built up a considerable reputation for the reliability of their TV sets and similar care will be taken to ensure that the computer continues this tradition.

In terms of software support, some 50 packages covering a wide range of applications will be available from the day of launch. In addition the choice of the Z80 processor will make translation of existing software a straightforward process. Add to this the CP/M compatibility and there should be no shortage of software.

As to whether the machine will sell or not, only time will tell. The price of £499 inclusive of disk drive is attractive but is it enough to attract customers? Allowing some £250 for the drive and interface gives a price of £250 for a 64 x 16K computer with a good, yet not spectacular specification. Good value when compared to the BBC micro but just how the Amstrad computer, again a Z80 machine with CP/M compatibility will affect the market is difficult to predict.

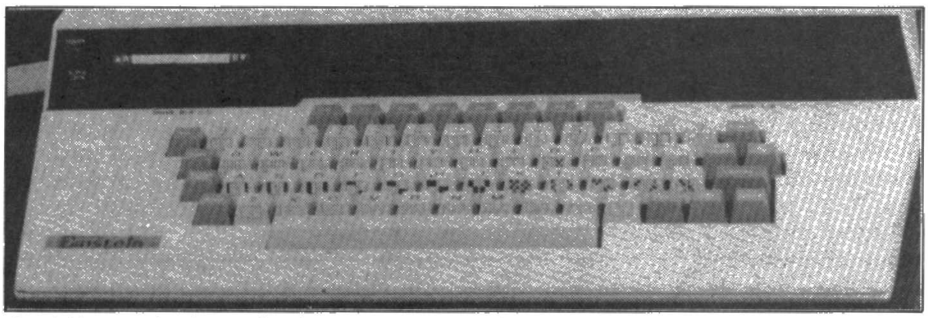
'The Einstein is a well-made computer'

Summary of assembler commands.

A	— arithmetic command – displays sum, difference and relative
B	— selects the baud rate for the RS232-C between 75 and 9600 bauds
C	— copies a block of memory from one location to another
D	— converts from hexadecimal to decimal notation
E	— execute command
F	— fill a block of memory
G	— go to specified location and execute the program there
H	— converts from decimal to hexadecimal notation
M	— modify memory contents
R	— read a block of data from disc
T	— tabulate memory
W	— write a disc
X	— "cold" start to program
Y	— "Warm" start to program
Z	— inspect Z80 registers

The Einstein's BASIC commands.

ABS	DIM	INCH	MUSIC	READ	SWAP
ADC	DIR	INCH$	NEW	REM	TAB
AND	DOKE	INP	NEXT	REN	TAN
APPEND	DOS	INPUT	NOT	RENUM	TCOL
ASC	DRAW	INPUT#	NULL	RESTORE	TEMPO
ATN	DRIVE	INT	OFF	RETURN	THEN
AUTO	ELLIPSE	IOM	ON	RIGHT$	TI$
BCOL	ELSE	KBD	OPEN	RND	TO
BEEP	END	KBD$	OR	RST	UNLOCK
BIB$	EOF	KEY	ORIGIN	RUN	UNPLOT
BTN	ERA	LEFT$	OUT	SAVE	VAL
CALL	ERR	LEN	PEEK	SCRN$	VDEEK
CHAIN	ERL	LET	PI	SEP	VDOKE
CHR$	EVAL	LIST	PLOT	SGN	VERIFY
CLEAR	EXP	LISTP	POINT	SHAPE	VOICE
CLOSE	FILL	LN	POKE	SIN	VPEEK
CLS	FMT	LOAD	POLY	SIZE	VPOKE
CONT	FN	LOCK	POP	SPC	WAIT
COS	FOR	LOG	POS	SPEED	WIDTH
CREATE	GCOL	MAG	PRINT	SPRITE	XOR
DATA	GOSUB	MGE	PRINT#	SPRITE OFF	ZONE
DEEK	GOTO	MID$	PROT	SQR	
DEF	HEX$	MOD	PSG	STEP	
DEG	HOLD	MON	PTR	STOP	
DEL	IF	MUL$	RAD	STR$	

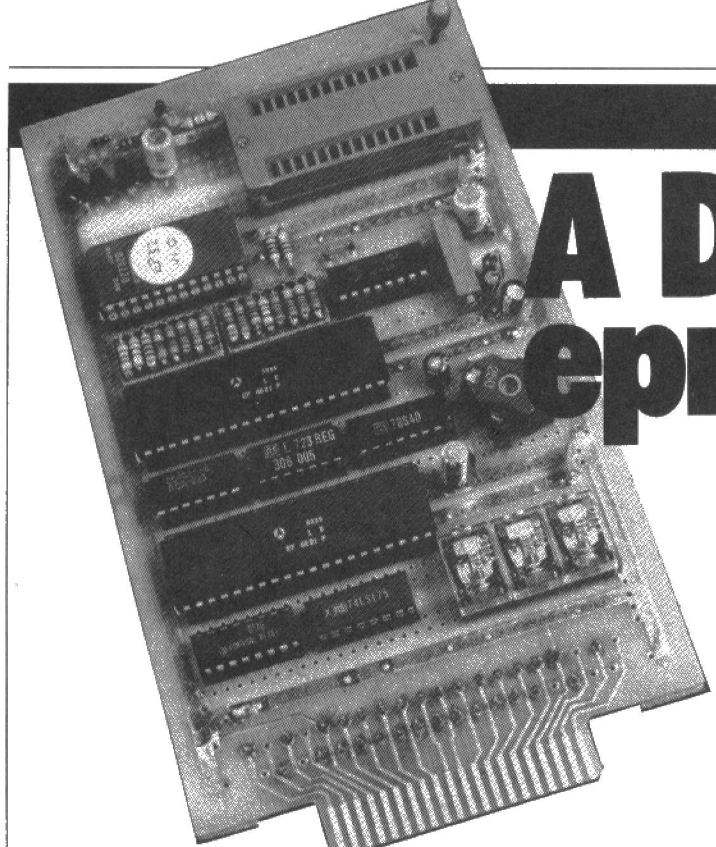


Einstein at a glance

Price	£499
Processor	Z80A running with 4MHz clock.
Memory	64K dynamic RAM with separate 16K video RAM.
Storage	3" 40 track compact floppy disk with 250K unformatted storage per side.
Display	40 x 24 and 32 x 24 text display. Resolution 256 x 192 pixels 32 Sprite planes.
Sound	Three tone plus white noise through internal speaker.
Keyboard	'Typewriter' style, full travel 67 keys including 48 alphanumeric 8 function and 11 control keys.
Expansion/ Interfaces	RS232, 8 bit user port, high speed 4 channel A/D converter, Centronics printer port, buffered bus.
Language and operating system	Tatung/Xtal DOS Tatung/Xtal BASIC 4

DOS command set.

DIR directory	GO-jump to 0100_H	PSW-password
DISP display	LOAD	REN-rename
DRIVE select drive	LOCK	SAVE
ERA erase	MOS-go to MOS	UNLOCK



A Dragon eprom blower

There is a lot of good utility software for the Dragon which would be easier to use on EPROM. Huw Jones has overcome the problem of the Dragon cartridge slot and built an EPROM board which fits snuggly inside it.

Inside the Dragon 32 lurks a 6809 microprocessor: probably one of the most important and appealing features of the machine for prospective users. The Dragon 32 is still the most cost effective introduction to the 6809, Motorola's advanced 8-bit processor. Any superlatives which may have been associated with a reference to the chip are justified and once you are familiar with 6809 assembly language programming the Z80 and 6502 seem very laborious indeed!

Amongst its main attributes are the logically structured instruction set, its internal 16-bit architecture permitting an extensive range of operations and an inherent ability to execute position independent code. It encourages the uses to develop a modular, neat programming style. On the hardware front, the Motorola quadrature system clock timing scheme enables the device to be readily interfaced to peripheral circuitry including discrete TTL and Intel type LSI components.

Unfortunately, despite the commercial availability of several 6809 software development packages, the full power of the device has not been available to the user who wishes to extend his or her horizons beyond the cream case and experiment with custom designed interfaces. The problem lies in the "black hole" on the right hand side of the Dragon, otherwise known as the cartridge slot. All the requisite bus signals are available at this point but the UECL type connector used hardly encourages Dragon users to produce applications articles like those that have appeared for the BBC micro within the pages of *E&CM* in months past. The connector is intended for PCB edge connector insertion only and just devising a means of mating a prototype circuit with the connector is a daunting task in itself.

The situation has changed, with the introduction of a prototyping board which provides a very convenient environment for building and evaluating peripheral circuits for the Dragon. With a view towards expanding the machine code development capabilities of my machine, I wanted to implement a reasonably comprehensive EPROM programming capability which I could then use to put "Alldream", the 6809 assembler/debugger I use, into non-volatile memory.

Specification

The finished design requires no external power source and merely plugs into the Dragon 32 cartridge port. It has resident operating software in firmware which consists of 4K of machine code, written entirely in assembly language. Alternatively, since it is written as position independent code, the program code could be loaded and run from tape anywhere within the Dragon's memory.

The board can handle six types of single rail EPROM: the Intel types 2716, 2732, 2732A, 2764 and the Texas 2516 (functionally equivalent to the 2716) and 2532. Each EPROM type is software selectable from the keyboard. It has a 'power on' indicator LED together with a 'programming' LED. The EPROM programmer provides a total of 12 functions which are menu driven for ease of use. A "user friendly" approach has been adopted throughout, whereby each stage of an operation will either prompt for a user response or display a meaningful message before proceeding. BASIC may be re-entered at any time with the unit installed. Only minor modifications should be required to convert the software to run on a Dragon 64 and a Tandy Colour Computer with Extended BASIC.

Operating instructions

Upon initialisation, the relevant device type must be selected by typing a single digit corresponding to one of the permissible types listed on the screen:

0	2716
1	2732
2	2516
3	2532
4	2732A
5	2764

Once accomplished the menu is displayed in full on the screen. Most operations apply to a reserved section of Dragon memory (referred to as the 'map') which is used to accumulate the eventual EPROM program data. Every function works within the currently active upper/lower address limits as displayed upon the screen. These default to the maximum address of the EPROM being used, and if otherwise changed, apply only for the next menu operation selected.

Each function is selected by number as requested by the bottom screen line prompt. The available functions are:

0 VERIFY

This checks the contents of the map with the EPROM inserted in the programming socket. A "passed" message is issued if the data tallies, otherwise "failed" appears on the screen together with the errant location and EPROM data. Pressing 'RETURN' continues the check whilst typing 'H' (for halt) exits back to menu code.

1 LOAD FROM EPROM

Data from an EPROM master in the ZIF socket is loaded into the 'map'. When finished, as with all the other operations, a message is displayed for 1 second before returning to the menu.

2 EXAMINE/CHANGE

This enables the map data to be modified. A starting address is asked for, after which the current data byte and address is displayed. Typing a hexadecimal digit will update the data byte from the right, eg $6A + "F" gives $AF. Note that only by pressing

'RETURN' will the 'map' contents actually update and move to the next sequential location. This helps prevent erroneous data alteration. The up/down arrow keys may be used to scroll through the map. An "H" resumes the menu.

3 EMPTY CHECK

Checks for a blank EPROM (all $FF's) in the programming socket.

4 PROGRAM EPROM

There is no automatic empty check before proceeding with programming; this is to enable selective overprogramming of already programmed devices. A 16-bit address indicator on the screen informs the user on how the programming is progressing. Where viable, a program verification is performed to establish the integrity of the programmed data. If it fails the operation is aborted after outputting an appropriate text message. For 32K devices where a program verify operation would involve switching Vpp between 5v and 25v/21v, a complete read verification is undertaken once programming has finished, notified by the string "completed copying, checking now". If a mismatch occurs then the user is informed and allowed to examine the bad locations as previously mentioned. To speed up the programming cycle, any *FF bytes cause an automatic skip to the next location, making the card very efficient for cases where only a fraction of the EPROM is actually required to be programmed.

5 CLEAR RAM

Sets the 'map' to $FF's.

6 COMPLEMENT RAM

Does an inversion (1's complement) operation on the 'map' data. This can often prove useful when generating custom inverse video CRT controller character generator EPROMs (bit of a mouthful!).

7 LIST

Dumps the current map contents to the screen or line printer. A start address is required in the range 0 to the upper limit of the EPROM type. If a hard copy is required then type "Y" in response to the issued prompt. If selected, the line printer gives a hex dump in 256 byte pages, of format 16 x 16 with column headings, row addresses and ASCII equivalents. Obviously the Dragon 32 screen format cannot accommodate this and so a 16 x 4 data display is adopted with ASCII equivalents. Pages may be scrolled up and down by using the relevant arrow keys. An "H" resumes the menu.

8 SET ADDRESS LIMITS

Allows the user to specify a restricted scope of addresses for the next operation; for instance, just to program $1C0-$1D9 inclusive for a program patch. Prompts are again issued.

9 LOAD FROM MEMORY

The map can be loaded with the contents of any chunk of memory within the 64K addressing range of the Dragon. The start load address which is requested may be any 16-bit value, eg the BASIC interpreter at $8000.

10 SET DEVICE TYPE

At any time, the target EPROM type may be changed. This permits, for example, the contents of two 2532's to be installed within a single 2764.

11 QUIT

Returns to the Basic interpreter. Re-enter by EXEC &HC002.

12 MOVE MEMORY BACK

With prompts, the user specifies a lower,

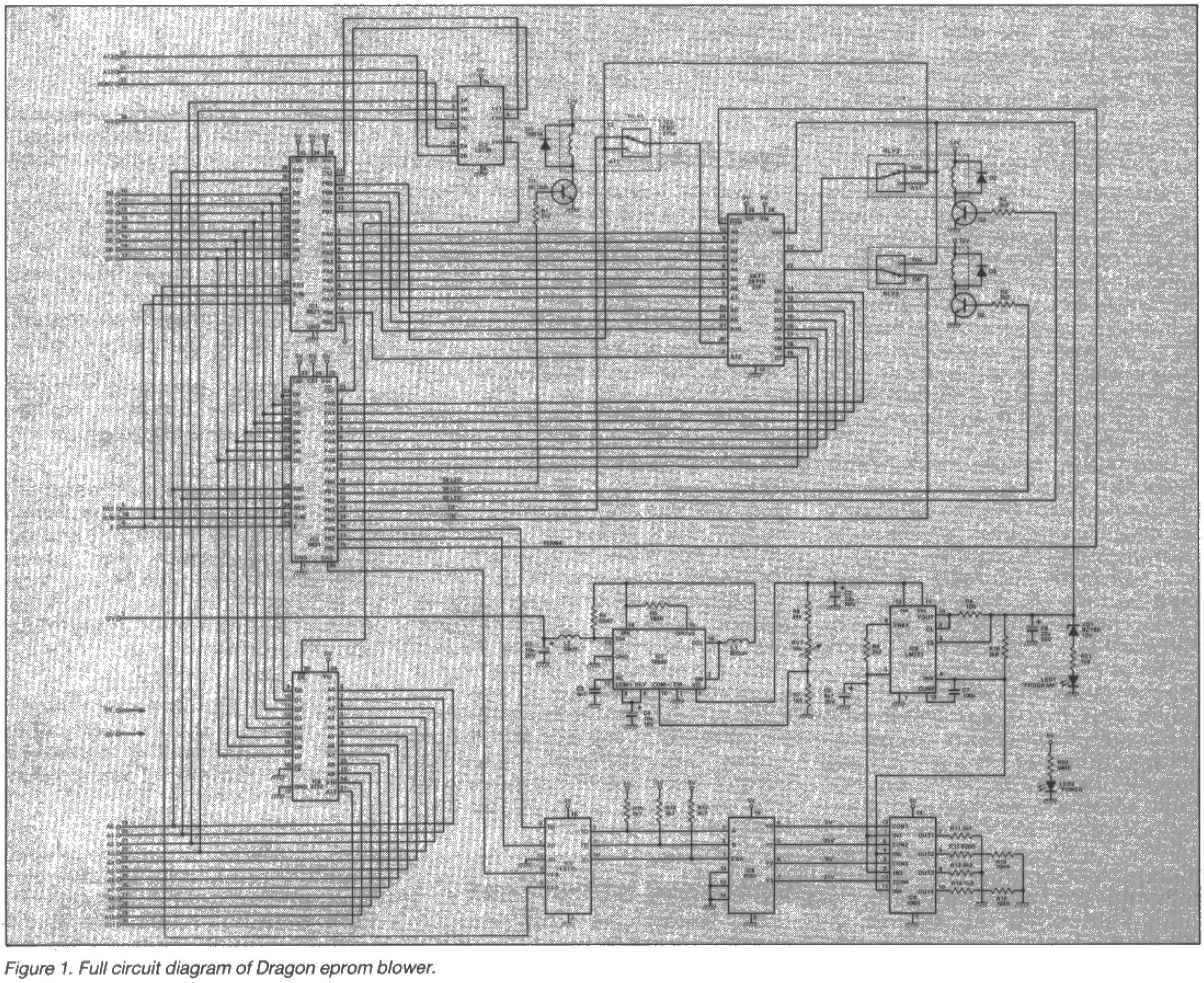


Figure 1. Full circuit diagram of Dragon eprom blower.

upper and destination address for a map data block movement.

Circuit Description

Two 6821 PIA chips are used to generate the ZIF program socket address and data lines, A0-12 and D0-7 respectively. In addition the CE and OE control signals are also software driven providing a great degree of flexibility. IC4, the 2732, is earmarked for the machine code operating software residing at $C000-CFFF. The address decoding is accomplished by an LS139, one half enabled by the IO2 signal from the Dragon 32 and the other section active when ROM2 goes low. These latter signals are derived internally from the RAM memory segment encoding section. Three single-pole changeover relays are used to alter the functions of the relevant socket pins to cope with the different device configurations.

Typically, an EPROM requires a fairly high voltage for programming to enforce the required movement of electrons onto the floating gate and thus change the state of the cell from a '1' to a '0'. In this application, the programming voltage required, Vpp, is either 25V or 21V according to the device being used. The necessity of providing an external power supply especially for this purpose is not just inconvenient but downright untidy.

A 78S40 switching regulator IC, produced by Fairchild, is used instead to generate the Vpp voltage by stepping up the Dragon's 12V rail. The diode charge pump principle is used, with L1 being the energy storage element. This is empowered to be a high Q coil of 803uH, wound as 58 turns of 0.315mm enamelled wire on an RM6,250 AL ferrite core (RS number 228-220). The coil can be formed very easily and a great deal of latitude has been allowed for in this respect in the value chosen for R4, the short-circuit protection resistor. A miniature encapsulated 33uH choke, L2, is used in conjunction with C1 as a simple low pass filter network to minimise switching noise fed back to the supply. A switching frequency of approximately 20KHz is used, as set by C3. An internal comparator adjusts the duty cycle of the oscillator to govern the amount of charge reaching the output smoothing capacitor C2. This error amplifier compares an attenuated version of the actual output to an internal 1.3V reference. A trimpot is included for adjustment of Vo.

The remaining components constitute a programmable Vpp voltage supply. The 'raw' 29V or so obtained from the 78S40 is regulated by an LM723 which can generate one of four possible voltages: 5V, 25V, 21V or 6V – the latter being required to implement Intel's intelligent programming algorithm (not utilised here). A 4066 analogue gate modifies the resistor chain ratios of the LM723 when any one of its channels is 'on'. The resistance values used for R11 to R14 may need to be slightly different from those shown to account for any variance in the ROM of the 4066 employed. A 4555 2 to 4 line demultiplexer ensures that only one 4066 channel can be active at any one moment and also reduces the number of output control lines required from IC2. To ensure that the circuit will always default to the safe 5V output condition, an LS175 latch is used, so that a system reset will clear it. An added safety aspect here is that any 6821 based voltage selection must be clocked through to the 4555 by software.

Two miniature LED indicators add the finishing touch. A green "power up" signal shows the presence of the 5V rail to the card whilst a red "programming on" LED will only illuminate if the Vpp pin has at least 20V applied to it during programming. Although not shown, it is recommended that 0.01uF ceramic decoupling capacitors are used.

Construction Details

The card is built on a purpose designed prototyping board. There are two versions of the board with different edge connectors; tinned copper or gold plated strips. The latter type is advised if the finished unit is likely to undergo frequent use. The circuit is constructed using a Roadrunner pen (which is similar to the Verowire system but easier to use) and glue fixing wiring combs. The only problem that may be experienced with wire wrapping is the length of the pins on the underside nearest the cartridge slot extrusion. All the edge connector tabs are numbered on the board and 0.1" vero pins used for through connections.

Sockets are used for all the ICs and the board is laid out as shown in the picture. Because they are quite compact, several pads on the solder side need to be cut and this is done using a standard stripboard cutting tool. Before proceeding further, L1 should be made. As previously stated, the coil winding process is not critical. Next insert any decoupling across the supply rails and wire up the supply rails, then proceed to wire in all the remaining connections.

When ready, start the testing/debugging sequence by checking the operation of the 78S40 circuit without inserting any more chips. If there are no problems then a high pitched whistle (the coil resonating) should be just audible and the green LED will be alight. The output voltage should be measured as being between 25V and 35V and then trimmed to 29V via VR1. Insert the other devices one at a time with the mains switched off, always checking that the Dragon operates correctly when turned back on.

If everything seems to tally a functional test of the card can be performed by running the short BASIC test program provided in **Listing 1.** This exercises the component sections of the EPROM programmer (an oscilloscope is useful for the 6821 square wave output tests). Any available 2732 (or even a blank socket!) can be used for the final memory reading check. Selecting a Vpp of 21V or 25V should illuminate the red "programming" LED in the penultimate test. For the 6821 input test use a 4k7 resistor to connect any one of the ZIF data lines (pins 11-13, 14-19) to 0V (pin 14).

Finally repeat the Vpp test whilst connecting a 1K resistor between pins 1 and 14 of the ZIF socket to check the load regulation of the VPP supply. No significant voltage difference should be observed from the no load conditions. The card is now ready for the software. Note that the EPROM programmer, just like a standard cartridge, should never be inserted or removed from the Dragon with power supplied. ■

EPROM PROGRAMMER KIT

For those who would prefer to construct the Eprom Programmer as a kit, Steve's Electronics Supplies Ltd., Castle Arcade, Cardiff, have arranged to produce a double sided PCB with a corresponding kit of parts.

Continued next month.

PARTS LIST

Semiconductors	
IC1,2	6821
IC3	LS139
IC4	2732
IC5	LS175
IC6	4555
IC7	78S40 Switching reg
IC8	4066
IC9	LM723 voltage reg
Q1,2,3	BC182L
Diodes	
D1,2,3	IN914
D4	miniature LED red
D5	miniature LED green
ZD1	BZY88 12V
Resistors	
R1,2,3,8	2K2
R4	OR47 1/2W
R5	180R
R6	24K
R7	1K3
R9	10R
R10	3K9
R11	5K1
R12	820R
R13	8K2
R14	1K3
R15,16,17	4K7
R18,19	100R
R20	560R
R21	1K8
VR1	10K Trimpot
Capacitors	
C1	10uF 25V elect radial
C2,6	22uF 50V elect radial
C3	4n7 ceramic plate
C4	10uF 16V elect radial
C5	4u7 25V elect radial
C7	100pF ceramic plate

L1: 803uH coil (see text); L2: 33uH encapsulated choke;RL1,2,3: FBR211A 12V; SK1: 28PIN ZIF socket; all resistors ¼W 1% unless stated.

SOFTWARE REVIEWS

ADE

BBC Model B
System Software £60.00

ADE is a 16K EPROM that offers a host of extremely useful features to anyone who writes extensive assembly language programs on the BBC Micro.

It consists of an assembler, with full macro facilities, an editor and a monitor/disassembler – a complete development package. It comes in a large box with a comprehensive manual and a demonstration disc. Each part of the ROM is interconnected, in that an assembler needs an editor to create its source files, and a monitor is needed to debug the resultant object code.

So, taking the editor first: this is based on the well-known TECO editor (TM of Digital Research) – which was a character editor – but this one is altered somewhat and comes across more as a screen editor. All the usual editing functions are there; character insert and delete, block move, insert and delete, search and replace, cursor movement, and excellent file handling. Like all of ADE, the editor is primarily intended for disc based systems, although it will work in a more limited fashion with cassette filing systems. The attitude taken is that no sensible programmer uses tapes – which is not as arrogant as it sounds.

The assembler comes next, and is notable for the fact that it assembles **from** disc **to** disc, making it very efficient and thus capable of assembling very large source files. This is aided by a CHN directive, which tags one source file onto the end of another as far as the assembler is concerned. In fact the number and scope of the assembler directives and pseudo-ops is gratifyingly large, and it rather makes the Basic assembler look silly.

Macros are supported, and in fact there is a macro library supplied on disc, which can be extended or depleted as required. Assembly listings are normally sent to the screen, but, of course, these can also be sent to the printer and/or a spooled file if required. Considering the disc-to-disc assembly, it is relatively fast, assembling 40K of source into 4K of object in about three minutes with a screen listing, and a little less without.

It uses the standard mnemonics and symbols, so that **S** rather than **&** signifies hex, but to aid compatibility with the Basic assembler and less intelligent users (!), either can be used.

The monitor is in a way the poor brother of the other functions, as it has been squeezed into the very small space left over. Nevertheless it offers all the functions one would expect, including single stepping through both RAM and ROM, but it is

SOFTWARE FILE

A BUYER'S GUIDE TO UTILITY SOFTWARE

Serious home computer users are demanding useful, and cheap utility software. A wealth of new products are available in this expanding market, and this month *E&CM* publishes a comprehensive buyer's guide to programmer's utilities. Forthcoming issues will contain guides to graphics packages, languages, word processors, communications etc.

MACHINE	PACKAGE	PRICE	FORMAT	MEMORY	SUPPLIER	COMMENT
ASSEMBLERS						
ATARI	ASSEMBLER EDITOR	39.99	CR	8K	ATARI	USES ACRONYMS
ATARI	MACRO ASSEMBLER	59.99	D	32K	ATARI	REVISES AND CHANGES GRAPHICS
CBM 64	ASSEMBLER DEVELOPMENT	25.00	D	64K	CBM	
DRAGON	ASSEMBLER EDITOR	7.00	C	32K	DYNATECH	AIDS MC PROGRAM WRITING
DRAGON	DASM	18.95	CR	32K	COMPUSENSE	TWO PASS ASSEMBLER (BASIC)
DRAGON	DASM/DEMON	30.45	CR	UX	COMPUSENSE	COMBINED MONITOR/ASSEMBLER
DRAGON	DISASSEMBLER	5.00	CR	32K	HILTON	MC DISASSEMBLER
DRAGON	DREAM	10.95	C	32K	DRAGON DATA	TOP QUALITY DRAGON ASSEMBLER
DRAGON	EDT/ASS	19.95	C	32K	MORRISON	GLOBAL ASSEMBLER INCLUDES MON AND DIS
DRAGON	ENCODER	29.95	CR	32K	PREMIER	ASS/DIS EDITOR
ORIC	ASS/DIS	6.95	CR	16K	DURELL	ALSO A 48K VERSION
ORIC	ASS/DIS	9.00	C	48K	SEVERN	ASS/DIS EDITOR. FULL 6502 MNEMONICS
ORIC	ORION ASS/DIS	12.95	C	UX	LOTHLORION	WITH FULL MON/DEBUGGING FACILITIES
SHARP MZ700	EDITOR/AS	16.50	C	—	KUMA	WITH FULL EDITOR AND OBJECT DEBUGGER
SPECTRUM	DEVPAC	14.00	C	16K	HISOFT	FAST, HIGHLY RECOMMENDED (*E&CM* APRIL 1983)
SPECTRUM	EDT/ASS	8.50	C	16K	PICTURESQUE	REVIEWED FEBRUARY 1984
SPECTRUM	INFRARED	6.75	CR	16K	ACS	DIS TO ULTRAVIOLET
SPECTRUM	PLUS 80	19.95	CR	48K	OXFORD COMPUTER	GIVES 80 COL CENTRONICS PRINTOUT
SPECTRUM	ULTRAVIOLET	7.50	CR	16K	ACS	ASSEMBLES ALL Z80 MNEMONICS
SPECTRUM	EDTR/ASS	12.50	CR	16K	KUMA	EDITOR/ASSEMBLER
SPECTRUM	ZEUS ASS	12.95	CR	48K	SINCLAIR	NOT THE BEST
SPECTRUM	M/C ASS	7.95	CR	16K	McGRAW HILL	
SPECTRUM	AM-EDIZON	10.00	C	16K	AMERSHAM	COMBINED ED/ASS
ZX81	ASSEMBLER	9.95	C	16K	ARTIC	
SPECTRUM	ULTRAVIOLET	7.50	C	16K	ACS	ALL DECIMAL – NO HEX
ZX81	ZXAS	6.95	C	16K	BUG BYTE	FULL SPEC Z80 ASSEMBLER
ZX81	AMAZON	8.00	C	16K	AMERSHAM	CREATES M/C FILES
BACK-UP TAPES/COPIERS						
BBC	SHADOW	8.00	C	32K	CLARES	COPIES 99% OF ALL KNOWN TAPES
BBC	REPLICA	9.95	D	32K	CLARES	UPLOADS CASSETTES ONTO DISC
BBC	GLONE	5.00	C	16K	WORK FORCE	UNLOCKS 'LOCKED' FILES
BBC	MASTER COPIER	6.50	D	16K	AZTEC	COPIES M/C TAPES
BBC	COPYFILE	5.00	C	16K	VISION	1.2 O/S T-T/T-D
CBM64	SUPER DISK UTILITY	12.95	D	UX	ADAMSOFT	SINGLE DRIVE 1541 BACKUP
CBM64	1541 BACKUP	10.00	C	64K	SUPERSOFT	BACK-UPS FOR 1541 drives
SPECTRUM	CLONE	5.00	C	16K	WORK FORCE	BASIC M/C COMPILER
SPECTRUM	CLONEIT/RENUMBER	5.99	C	16K	ARGUS	COMPIER/RENUMBER FACILITIES
SPECTRUM	THE SAFE	5.50	C	48K	PEEK SOFTWARE	BACK-UP FOR HOME TAPES
SPECTRUM	TAPE CONTROLLER	NA	C	16K	LERM	BACK-UP FOR HOME TAPES
SPECTRUM	ZAP	6.00	C	16K	SCIMITAR	HANDLES 16K AND 48K M/C
VIC 20	TAPE BACKUP	7.95	C	UX	LEVEL	BACKS-UP MOST BASIC/MC
COMPILERS						
BBC	TURBO COMPILER	9.95	C	16K	SALAMANDER	M/C BASIC COMPILER
CBM64	PETSPEED	50.00	D	UX	CBM	SPEEDS UP EXECUTION OF BASIC PROGRAMS

rather let down by its handling of breakpoints. Although – as in most monitors – these simply consist of BRK (00) instructions, a breakpoint table is not maintained, so that a user would have to remember the replaced instruction(s). Not only that, but the program counter and memory pointer end up after the BRK, so that you have to physically move back to continue. This is only a relatively minor fault, but it can cause annoyance to those who are likely to use this facility a lot. Like me.

The rest of the monitor consists of a standard front panel display and a host of commands for altering, moving and checking memory. There is also, of course, a disassembler.

ADE is a very good purchase with only very minor faults, and it is hard not to recommend it to all 6502 programmers.

£60 is a lot of money, especially when one considers that you would need to purchase System's SPY2 ROM as well to get a real monitor, but in the long run there is no doubt that it is worth it. **AD**

SPY2

BBC Model B
System Software £24.15

This 8K EPROM is an extension of the SPY monitor contained in ADE (see above), which was available separately in EPROM as SPY. SYSTEM quickly noticed that the original version, although reasonable, could be an awful lot better, so they amended it, added a few *Disc Doctor* like functions, and released it.

The front panel is obtained by typing * SPY, but as this may conflict with ADE, * DEBUG has the same effect. This selects and clears a Mode 7 screen, and produces a standard front panel containing register contents and a memory pointer plus the contents of the surrounding memory. The colours chosen are fine, and can be improved for those with black and white displays by pressing W.

Register and memory contents can be altered, moved, checked and deleted, as you'd expect, and programs can be single stepped. It is nice to discover that subroutines can be treated as one single instruction during single step.

The single step facility can also be used as an automatic trace with dump to printer; all operating system calls are labelled as in the BBC Micro User Guide, which is very useful. This feature actually occurs in all the disassemblies the ROM creates.

The breakpoint handling is much improved over the original SPY, with a dynamic table maintained so that programs or routines can be run up to a predefined point (or points), registers and so on checked, and then continued up to the next point. A delight to use.

Rather interesting are the various disassembly options that are provided. The standard disassembly invocation results in an output similar to other programs, with

MACHINE	PACKAGE	PRICE	FORMAT	MEMORY	SUPPLIER	COMMENT
CBM64	TINY BASIC COMPILER	9.95	D/C	UX	ADAMSOFT	COMPILES BASIC SUBSET TO TRUE 6502 M/C
SHARP MZ700	PASCAL COMPILER	39.50	C	—	KUMA	CONFORMS TO STANDARD PASCAL
SPECTRUM	COMPILER SUPER C	9.95	C	—	SOFTEK	ONE OF THE BEST
SPECTRUM	MCODER II	9.95	C	48K	PSS	TRUE INTEGER COMPILER
SPECTRUM	IS	9.95	C	16K/48K	SOFTEK	COMPILER 17K BASIC
SPECTRUM	FP	19.95	C	16K	SOFTEK	INC. FLOATING POINT HANDLING
VIC 20	TINY BASIC COMPILER	9.95	C/D	3K	ADAMSOFT	HANDLES FP
ZX81	MCODER II	8.95	C	16K	PSS	INTEGER COMPILER
DATA COMPRESSION						
CBM64	COMPACTOR	NA	C	64K	SUPERSOFT	REMOVES REM's, SPACES ETC.
DEBUGGERS						
BBC	BEEBMON	8.00	D	16K	MICROPOWER	M/C DEBUGGER
BBC	DEBUGGING PROGRAM	NA	C/R	48K	MICRO-AID	WORKS ALONGSIDE THE BBC's BUILT-IN ASSEMBLER
BBC	DEBUGGING	10.30	C	32K	ADDISON-W	USED ALONGSIDE BOOK, 'ASSEMBLY LANGUAGE PROGRAMMING'
BBC	GREMLIN	33.35	R	—	COMPUTER CONCEPTS	M/C MONITOR ROM
SPECTRUM	M/C TEST TOOL	9.95	C	16K	OXFORD COMP PUBS	COMPLETE WITH TUTORIAL COURSE
DIAGNOSTICS						
SPECTRUM	MICRODOCTOR	7.00	C	16K	J. K. GOSDEN	CHECKS CPU, RAM, PRINTER AND SPEAKER
DISK/FILE HANDLING						
BBC	DISK DOCTOR	NA	R	48K	MICRO AID	NOT TO BE CONFUSED WITH COMPUTER CONCEPTS VERSION
BBC	DISK FILE SYSTEM	30.00	D	48K	MOLIMERX	ALLOWS FOR 63 FILE NAMES AND 15 CHARACTERS
BBC	APPEND IT	3.00	C/D	16K	AZTEC	PUTS PART PROGRAMS TOGETHER
BBC	INMENU	8.00	D	32K	MUSE	COPIES ITSELF AND BUILDS BOOT FILE DOCUMENTATION
BBC	CATSWOP	10.00	D	32K	MUSE	DOUBLES CATALOGUE SIZE FOR BBC
BBC	DISC-OVERY	20.70	D	32K	MOLIMERX	CONSISTS OF THREE SUB PROGRAMS
BBC	DISK ZAP	18.10	D	32K	MOLIMERX	STANDARD ZAPPING PROGRAM
BBC	THE KEY	12.95	D	32K	CLARES	EDIT AND RETRIEVE, FORMATTING BACK-UP
BBC	DISC DOCTOR	33.85	R	—	COMPUTER CONCEPTS	VERY USEFUL RANGE OF DISK COMMANDS
SPECTRUM	ZX SLOW LOADER	10.00	D	16K	E LONDON ROBOTICS	LOADS ZX81 PROGRAMS ONTO SPECTRUM
SPECTRUM	TAPE INDEXER	5.00	C	48K	WORK FORCE	PRINTS OUT HEADERS, ADDRESSES
ZX81	FAST LOADER	8.50	C	16K	JRS	SPEEDS RATE OF PROGRAM LOADING BY 10-15 TIMES
EDITORS						
COMM 64	EDITOR-PAR	35.00	D/C	64K	IMPEX	EDITING, PROGRAMMING, DEBUGGING TOOLS
DRAGON	EDIT PLUS	34.50	CR	32K	COMPUSENSE	51 x 24 SCREEN EDITOR
ZX81	AM-ZXEDIT	4.00	C	1K	AMERSHAM	EDITS SOURCE CODE UNDER AM-ZXMON O/S
EXTENSIONS						
CBM64	UPGRADER	74.75	C	64K	—	ADDS 50 COMMANDS
CBM64	POWER 64	69.99	D	64K	KOBRA	27 COMMANDS TO MAKE PROGRAM WRITING EASIER
CBM64	SIMONS BASIC	43.00	CR	64K	COMMODORE	RIGHTLY FAMOUS EXTENSION – ADDS 114 EXTRA COMMANDS TO LIMITED CBM BASIC
CBM64	VICTREE	65.00	CR	64K	SUPERSOFT	ADDS 40 COMMANDS
CBM64	ULTRABASIC	22.95	C/D	UX	ADAMSOFT	ADDS 50 COMMANDS
DRAGON	TOOLKIT	29.95	CR	32K	PREMIER MICROSYSTEMS	ADDS 60 COMMANDS
ORIC	EXTENDED BASIC	7.50	C	48K	SEVERN	13 EXTRA COMMANDS AND 2K
SHARP MZ700	TOOLKIT	14.50	C	UX	KUMA	ADDS FACILITIES LIKE COPY, PUT, TONE ETC.
SPECTRUM	EXTENDED BASIC	9.95	C	48K	CP SOFTWARE	10 ADDITIONAL BASIC COMMANDS
VIC 20	EXPANDER	34.95	CR	UX	CBM	MEM EXPANSION WITH NEW GRAPHICS, MUSIC ETC
VIC 20	BUTI PLUS	39.95	CR	UX	AUDIOGENIC	17 COMMANDS AND 3K MEMORY EXPANSION
ZX81	TOOLKIT	5.95	C	16K	ARTIC	9 NEW FUNCTIONS
MONITORS						
BBC	EXMON	10.00	C	16K	BEEBUG	REVIEWED *E&CM* MAY 1984
BBC	MICROMON	16.10	D	32K	MOLIMERX	DEBUGGER/MONITOR PROGRAM
BBC	MONITOR	13.80	C	32K	MOLIMERX	COMPREHENSIVE ERROR CHECKING
CBM64	MONITOR	50.00	C	64K	AUDIOGENIC	M/C MONITOR
CBM64	ZOOM	10.00	CR	64K	SUPERSOFT	SINGLE STEP TRACE ETC
DRAGON	DRABUG	12.95	C	32K	PSS	M/C MONITOR/DISASSEMBLER
DRAGON	DEMON	19.95	CR	32K	COMPUSENSE	M/C MONITOR WITH 13 COMMANDS
DRAGON	DREAMBUG	7.95	C	32K	DRAGON DATA	MON/DIS COMPLEMENTS 'DREAM' PACKAGE
ORIC	ORICMON	8.95	C	16K	PSS	MONITOR/DISASSEMBLER
ORIC	ORIC-MAN	15.00	C	48K	TANSOFT	

SPECTRUM	SPECTRAMON	5.99	C	16K	ARGUS	MONITOR UTILITY
SPECTRUM	MON/DISS	12.95	—	16K	SINCLAIR	
SPECTRUM	MONITOR	7.50	—	16K	PICTURESQUE	EFFICIENT BUT LIMITED
VIC 20	M/C MONITOR	14.95	C	UX	CRYSTAL	
VIC 20	M/C MONITOR	34.95	C	UX	CBM	18 COMMANDS FOR DEBUGGING
VIC 20	MONITOR	19.95	CR	UX	AUDIOGENIC	17 BASIC PROGRAMMING AIDS
ZX81	INFRARED	6.75	C	16K	ACS	M/C PROGRAMS READ AS Z80 MNEMONICS
ZX81	MON/DIS	8.95	C	16K	CRYSTAL	DISASSEMBLER/DEBUGGER IN ONE
ZX81	REMLOAD	7.00	C	16K	PICTURESQUE	BREAKPOINT AND REGISTER DISPLAY
DISK OPERATING SYSTEMS						
DRAGON	OS9	39.95	D	32K	DRAGON DATA	SEE REVIEW THIS MONTH
PROGRAMMING AIDS						
BBC	6502 DEVELOPMENT	49.85	D	32K	ACORNSOFT	INCLUDES LOOPING, RECURSIVE, CALLS AND SCREEN EDITOR
BBC	CARETAKER	33.35	R	—	COMPUTER CONCEPTS	17 PROGRAMMING UTILITIES
BBC	PROCAID	3.00	C	16K	MICRO-AID	INCLUDES 7 UTILITY PROGRAMS
BBC	PROCUAR	2.00	C	16K	MICRO-AID	M/C APPENDED TO BASIC
BBC	TOOLKIT	27.00	C	—	BEEBUG	REVIEW *E&CM* MARCH 1984
BBC	UTILITIES	5.95	C	32K	GOLEM	INC 1541 BACK-UP AND DOS WEDGE
CBM64	PROGRAMMER	14.99	D	NA	CBM	INC DISC BACK-UP, M/C MONITOR
CBM64	RENUMBER	7.00	C	64K	SUPERSOFT	RENUMBERS BLOCKS
COMM 64	TAPE/MERGE	7.00	C	64K	SUPERSOFT	MERGES CASSETTE WITH MEMORY
DRAGON	ALLDREAM	24.95	CR	32K	DRAGON DATA	ASS/DIS/ED/MON IN ONE PACKAGE
DRAGON	TOOLKIT	NA	C/R	NA	ASHBY COMP	EXAMINE, SEARCH, TRACE ETC.
DRAGON	UTILITIES	5.00	C	32K	HILTON	3 ROUTINES, COMPRESS, COPY CATALOGUE
SPECTRUM	SUPERCODE	9.95	C	16K	CP SOFTWARE	100 ROUTINE M/C TOOLKIT
SPECTRUM	MASTER KEY	5.95	C	16K	SOFTEK	EXECUTE UP TO 100 BYTES BASIC WITH ONE KEY PRESS
SPECTRUM	COMPLETE PROGRAMMER	9.95	C	16K	SOFTEK	ED/MON/ASS/DIS
SPECTRUM	M/C PROGRAMMER	12.95	C	16K	SOFTEK	ED/MON/ASS/DIS
SPECTRUM	MASTER TOOLKIT	9.95	C	16K	OXFORD COMPT PUBS	28 COMMANDS
SPECTRUM	PROGRAMMER DREAM	6.95	C	16K	WORK FORCE	RELOCATABLE CODE
SPECTRUM	ENHANCE YOUR BASIC	7.00	C	16K	MALVA	BASIC TOOLKIT
SPECTRUM	TTS	7.95	C	16K	TIMEDATA	5 TOOLKIT PROGRAMS
SPECTRUM	SUPER TOOLKIT	NA	C	16K	NECTARINE	TRACE, RENUMBER, COMPRESSION
VIC 20	PROGRAMMER AID	34.95	CR	UX	CBM	BEGINNERS ONLY
VIC 20	MINIKIT	8.00	C	5K	AUDIOGENIC	COMMANDS AND MEMORY EXPANSION
VIC 20	SUPERSCREEN	10.00	C	8K	AUDIOGENIC	40 COL DISPLAY
ZX81	TOOLKIT	5.95	C	16K	ARTIC	PROVIDES 9 NEW FUNCTIONS
ZX81	M/C TEST TOOL	9.95	C	12K	OXFORD COMPUTING	ENTERS AND TESTS HEX
ZX81	SCREEN KIT	5.70	C	16K	PICTURESQUE	SCROLLS SCREEN IN 4 DIRECTIONS
PROGRAM CONVERTERS						
BBC	ATOM LOAD	10.00	C	16K	ECCE	LOADS ATOM TAPES TO BBC
CBM64	PAL64	79.00	D	UX	KOBRA	CONVERTS TO 6500 M/C
DRAGON	DECODE	18.95	CR	32K	COMPUSENSE	TANDY BASIC TO DRAGON FORMAT
DRAGON	CRUNCHER	8.00	C	32K	—	DRAGON TO TANDY COLOUR CONVERTER
SPECTRUM	AM-ZXSP	7.00	C	16K	AMERSHAM	LOADS ZX81 TO SPECTRUM
SPECTRUM	SLOWLOADER	10.00	C	16K	E LONDON ROBOTICS	ENABLES SPECTRUM TO LOAD ZX81 PROGRAMS
PROGRAM GENERATORS						
ATARI	CODEWRITER	85.00	D	48K	DYNATECH	PRODUCES STAND ALONE PROGRAMS
BBC	ADVENTURE COMPILER	8.75	C	32K	MOLIMERX	ADVENTURE PROGRAM GENERATOR
SPECTRUM	THE QUILL	NA	C	48K	GILSOFT	ADVENTURE PROGRAM GENERATOR
SECURITY						
CBM64	PALGUARD	350.00	C/D	64K	SOFTGUARD	SECURES AGAINST PIRACY
SHARP MZ2700	FIENDISH ENIGMA	9.95	—	—	KUMA	A PRIZE IF YOU CAN CRACK THE CODE
SUB-ROUTINE LIBRARIES						
BBC	PROCFLUSH	1.00	C	16K	MICROAID	CLEARS OUT VARIABLES THAT OPERATING SYSTEM DOES NOT CLEAR WHEN RUN
SPECTRUM	M/C UTILITIES	5.00	C	48K	OED	SUITE OF M/C ROUTINES
SPECTRUM	SUPERCODE	7.00	C	16K	CP SOFT	M/C TOOLKIT WITH 60 PLUS ROUTINES
ZX81	PEP	6.00	C	16K	R&R	M/C ROUTINES TO TRANSFORM BASIC PROGRAMMING

data interpreted as instructions unless they don't make sense, but SPY2 also features an intelligent disassembler that keeps a record of all data references that fall within the range given, and outputs this as defined data bytes rather than 6502 mnemonics. Naturally this is not totally foolproof, but it is very handy and a very rare thing to find on such a small and comprehensive ROM. I would venture to say that SPY2 is the best BBC monitor I've seen.

The there are the disc utilities, ranging from simple disc sector editing to downloaders, formatting programs, catalogue editors and string search. In fact this ROM offers almost as much as *Disc Doctor* as far as discs are concerned, but is almost £10 cheaper.

Very definitely an essential purchase.

ADE and SPY can be obtained from System, 12 Collegiate Crescent, Sheffield.

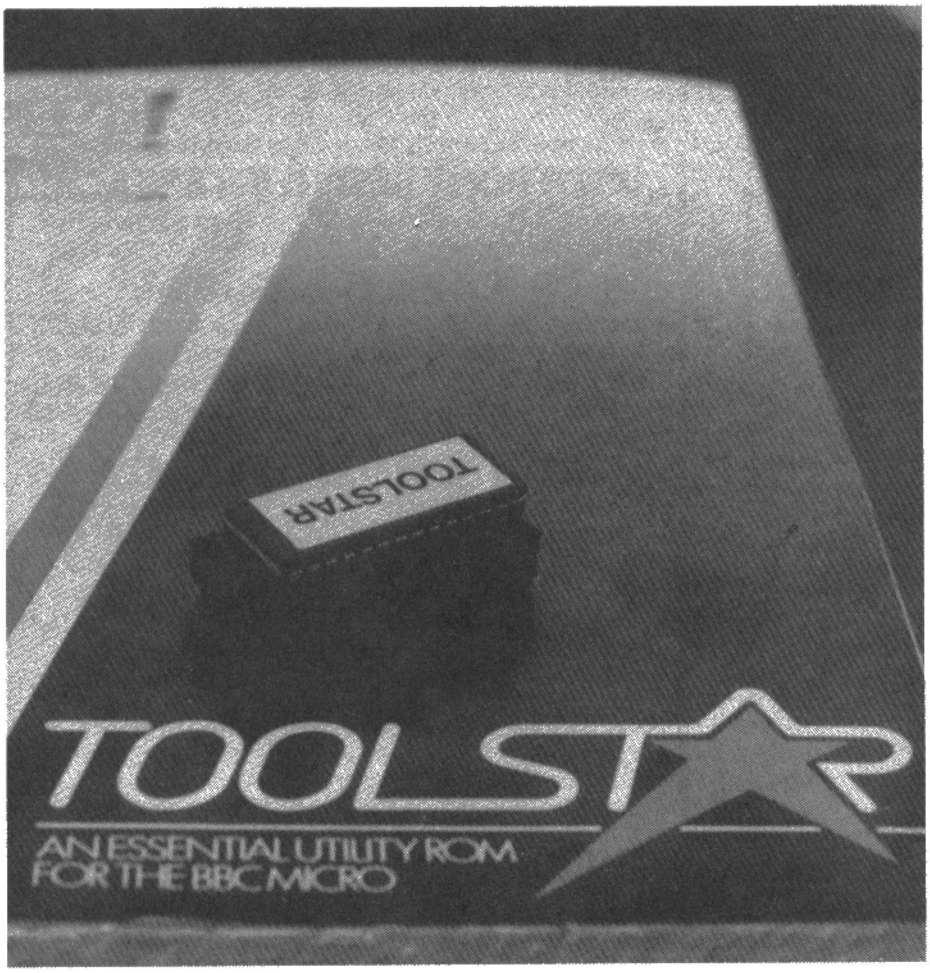
AD

SUPPLIERS

Acornsoft	0223 316039
ACS	0532 667440
Adamsoft	0706 524304
Addison Wesley	01 631 1636
Amersham Software	02403 6231
Argus	01 437 0626
Artic	0401 43553
Ashby Computer Centre	
Andivgenic	0734 586334
Aztec	0742 862246
Beebugsoft	0727 54280
Bugbyte	051 7097071
CBM	0753 79292
Claves	0606 48511
Compusense	01 882 0681
Computer Concepts	
CP Software	02406 3783
Crystal	
Dragon Data	0656 744700
Durell	0823 54489
Dynatedn	0481 20155
ECCE Productions	01 302 1667
East London Robotics	01 474 4430
English Software	061 835 1358
Gem	0279 723518
Gilsoft	
Golem	0344 50720
J K Gosden	
Hilton	0689 35101
Hisoft	0296 688995
Impex	01 900 0999
JRS	0903 65691
Kotra	0491 572512
Kuma	07357 4335
Lerm	
Level	021 643 6728
Lothlorian	0625 876642
Malva	
McGraw Hill	0628 23432
Micro-Aid	0209 831 274
Micropower	0532 683 188
Molimerx	0424 223 636
Morrison	0532 480987
Myrmidon	073784 2072
Nectarine	
Oxford Computing	0753 888866
Peak	0283 44904
Picturesque	01 777 0372
Premier Microsystems	01 659 7131
PSS	
Quicksilva	0703 20169
Salamander	0273 771 942
Scimitar	
Severn	0594 43352
Sinclair	0276 685311
Softguard	
Softek	01 240 1422
Supersoft	01 861 1166
Tansoft	02205 2261
Tasman	0532 438301
Timedata	
Tomorrows World	06845 62467
Vision	01 748 9009
Workforce	

Toolstar

Vincent Fojut reviews a useful addition to the arsenal of BBC machine-code and BASIC utility ROMs.



Toolstar Utility ROM

Pace, £34.00 inc. VAT

A new addition to the bewildering array of dedicated ROMs for the Beeb is Toolstar from Pace, a company best known for its Amcom Disc Filing System.

The 8K ROM comprises a range of utilities which can be split into three main groups: machine code tools, BASIC tools and disc filing system tools. To see the range of commands available, complete with the appropriate syntax, there are three menu-names which can be invoked with the *HELP command.

Table 1 shows the result of typing *HELP MCODE. In common with BBC OS calls, all Toolstar commands are preceded by an asterisk, "*". For convenience, all machine-code utilities start with an "M". The function of each command is as follows:

MDUMP – This is a hex/ASCII dump in the now standard format, each line comprising a hex address; eight hex data bytes, followed by their ASCII equivalents. Non-printable ASCII characters are displayed as a dash (–). Only one line is printed at a time and the listing is continued by holding down the Return key. Although one-line printing may appear inconvenient, it simplifies the use of *MDUMP from BASIC programs, where listing can be tailored to specific demands.

MDIS – An above-average disassembler. Features of note are that operating system calls within the range &FF00 to &FFFF and their indirection vectors are printed by name. Regrettably, the disassembler cannot recognise data blocks within machine code. However, the routine does print ASCII equivalents of the hex bytes being disassembled, so ASCII strings, at least, are relatively easy to detect, even if the "instructions" interpreted alongside are not true operations. Another interesting facet of both MDUMP and MDIS is that output can be directly edited, using the cursor and copy keys. Replacement bytes can be entered in either hex or ASCII.

M – This single character command automatically enters "memory editing" mode, defaulting to whichever format was last selected (either MDUMP or MDIS).

MFIND – Performs a complete memory search for a given byte, or sequence of bytes. In common with most commands, the bytes may be entered as two digit hex numbers, or ASCII characters preceded by a period, ie 41 or .A would both represent a byte of decimal value 65. A series of bytes can be entered as a mixture of both formats. Conveniently, an additional command (see MROM later) allows any resident paged ROM to be selected for inclusion in the memory search performed by MFIND.

MSEED – This command places a given value into every byte within a specified range of memory. This can be useful for setting memory to a known value before running a program which modifies the area in some way.

MCOPY – A "block move" by another name. It can move data either up or down in memory, and correctly handles overlapping areas of "source" and "destination" blocks – that is, no valuable data is overwritten!

REPLACE – This command searches for a given string and replaces it with another. By default, the replacement is on a selective, not global, basis. That is, as each individual occurrence of the search string is found, the routine asks if you wish to delete the string, or just go straight on to the next occurrence. I was pleased to see that the routine accepted a "replacement string" of zero length, thereby doubling as a "delete string" command. All the same, I did miss the option of a "global" replace facility.

FIX – If "Bad Program" messages are driving you mad, the FIX command will almost certainly be of help. Although it does not cure errors, FIX at least renders your programs "listable". It does this by ensuring all line-length bytes match up with corresponding carriage-return codes which mark the end of each line. However it does not work if the very beginning or end of the program are missing. Fortunately, this is one

command that I haven't had to use yet (now, if that isn't tempting fate, what is?).

FLIST – Performs a "Forced Listing" of BASIC programs. This resembles a normal BASIC listing, but also prints the start address of each line in memory, plus the length byte, at the beginning of each BASIC line. A useful tool for investigating the causes of "Bad program" messages on any afflicted program files.

RESET – A "very hard" reset! Invoking this command can be considered roughly equivalent to turning the machine off, then on again. All ROM areas used by BASIC are wiped down, and the operating system initialisation routine is then called. In view of the drastic nature of the routine, the message "Are you sure Y/N?" appears whenever the command is executed. To be honest, this is another utility that I never found the need to use.

FKEY – Another utility which is becoming a bit of an old chestnut – a function key viewer/editor. In this implementation, embedded control characters are displayed as spaces (but they copy normally using the Copy key). The effect of these control codes can be deduced, to some extent, by the appearance or colour of subsequent strings, but even with FKEY, there is not always an obvious link between the actual characters entered and the function definition displayed.

The remaining commands listed in the *HELP TOOL menu all relate to the Disc Operating System. There are four low-level utilities:-

DFORMAT – Formats and verifies a disc to Acorn standard.

DVERIFY – Verifies all or part of a disc.

DLOAD – Loads a disc sector into a specified memory area.

DSAVE – Writes a disc sector from a specified memory area.

Although simple, these commands can be used as "building-blocks" for more ambitious utilities, by embedding them within BASIC programs. A whole chapter of the manual is devoted to this. A number of programs are included, as an example of the kind of development possible – for example, a program to screen-dump a disc catalogue in hex and ASCII, by using DLOAD in conjuction with MDUMP.

The "expansion" philosophy not only applies to using existing routines within higher-level programs, but also applies equally to the development of low-level machine-code utilities to sit side-by-side with Toolstar's own vocabulary. By writing your own routines, stored in RAM, and supplying the start address of the routine to Toolstar using the *EXTEND command, you can use your own routines just like any other Toolstar command. User-supplied tools have their own menu available to them, accessed by entering *HELP EXTEND. Any user-defined code must adhere to strict convention, which are clearly laid out in a 14-page chapter of the manual, "Extending the Toolkit".

MCOMP – Executes a byte-for-byte comparison of two areas of memory, printing out all the addresses where a mismatch is found. Not only is it useful for verifying correct transfer after MCOPY, but it is particularly handy for comparing two similar versions of a program (for example, when you can't remember which is the latest!)

MROM – As mentioned briefly above, MROM defines which paged ROM is to be examined by the appropriate Toolstar commands. Once a given paged ROM is selected, then any Toolstar commands prefixed by M (ie the machine-code tools) consider that ROM to be part of current memory. The manual claims that extra ROMs on extension boards are also handled correctly by the command.

MBRK – A breakpoint handler, for debugging machine-code programs. By inserting BRK instructions at desired points within machine-code, and then activating the MBRK facility, the user can obtain a display of the 6502's internal register contents at the point at which the BRK instruction is encountered. In addition, the current paged ROM selected by MROM is shown. MBRK can also be used to intercept the standard error handler employed by the OS and BASIC interpreter.

MCRC – This final "machine code tool" performs a true cyclic redundancy check, returning a unique value, or "signature", for a section of memory. It is a useful means of checking the integrity of an important piece of code after loading. However, whilst it brings an discrepancy to your attention, it does not tell you exactly where the error lies.

TABLE 1. Machine-code tools.

>*HELP MCODE	
Toolstar 1.77	
MDUMP	<start> (,end) (bytes) (')
MDIS	<start> (,end) (bytes) (')
MFIND	<byte> (byte/s)
MSEED	<start> <end> <byte>
MCOPY	<start> <end> <dest>
MCOMP	<start> <end> <dest>
MROM	<rom no>
MBRK	<0/1> OFF/ON
MCRC	<start> <end>
M	editor

TABLE 2. BASIC and DFS tools.

Toolstar 1.77	
FKEY	(number)
OPEN	<line no>,(line no)
CONV	<dec> OR <&hex> OR <%bin>
FIND	<string>
REPLACE	<string>,<string>
FIX	
FLIST	(start)
RESET	
DLOAD	<drv>,<trk>,<sec>, <adds>
DSAVE	<drv>,<trk>,<sec>, <adds>
DVERIFY	<drv>,<total trks>
DFORMAT	<drv>,<total trks>

So much for the machine code extensions. A list of BASIC and Disc Filing System utilities can be had by typing *HELP TOOLS. All of the BASIC-related commands operate by default on the area of memory delineated by PAGE and TOP. There are eight such commands:

CONV – Given an input number in decimal, hex (preceded by &) or binary (preceded by %), CONV prints the number in all three forms. I found this command of limited usefulness, since hex/decimal/hex conversion is already so easy on the Beeb. Binary conversion can be handy for converting logical "bit-masks", for use by the AND, ORA & EOR instructions in the 6502 assembler. However, if you are in the middle of entering an assembly source program, it is quite a disincentive to have to exit and re-enter BASIC, just to be able to use CONV.

OPEN – A selective "smart" RENUMBER. OPEN allows you to specify any line in your program as the starting point of the RENUMBER operation. All forward and backward references in GOTOs, GOSUBSs and the like are automatically taken care of. But, as with the standard RENUMBER, there is no way to cope with computed GOTOs, such as:

GOTO (X * 3)

I could not really see much advantage in this command. I have yet to come across a truly comprehensive RENUMBER utility for the Beeb which does ALL I would like – for example, taking any block of line numbers and replacing them (or copying them) anywhere else within a BASIC program.

FIND – An old faithful, but an extremely useful command. FIND lists all lines in a BASIC program which contain a specified string. The command can search for ASCII strings (complete with wildcard, or "don't care", character facility), BASIC reserved words in tokenised format, or a combination of both.

The comprehensive manual is a thorough, well-balanced piece of work. A couple of oversights have been made (for example, who wants a dissassembler 'enhancement' which analyses memory in 8-byte steps?) but fortunately, these are the exception rather than the rule. There are concise introductions to machine code and the structure of BBC BASIC, with helpful hints and programs for more sophisticated needs. Overall, the impression is one of care and attention, from the clear descriptions of each command, right down to little touches like different coloured pages for each logical division.

At £34.00, Toolstar is not the cheapest utility available, but given the range of commands, their inherent power, and ease of enhancement, Toolstar compares very favourably with similarly priced utility ROMs. I found it very easy to use, and almost indispensable for any low-level work. I would consider it a useful addition to the arsenal of machine-code and Basic Programmers alike. ■

BOOK REVIEWS

Harry Fairhead's monthly guide

Word Processing for Beginners
by Susan Curran
Granada, 1984

Lots of books have already been published on office automation and "professional" word processors but this is the first that I have come across which talks about the sort of word processing that I have personal experience of – namely running limited word processing packages on fairly humble micros.

The book opens with a short and completely non technical explanation of what word processors do, concentrating upon the advantages of a computer over the traditional use of the typewriter, rather than on the intricacies of text processing. In Chapter 2 (on "Assessing your Needs") Susan Curran poses some sensible questions and draws attention to the different needs of accomplished, as opposed to inexpert, typists and to people who will use their computers for a variety of applications – the occasional letter writer as compared to a journalist or author. Chapters 3 and 4 examine hardware.

The author has some sound advice about technical points to consider when purchasing a printer. She alerts the reader to features such as print quality, paper width, speed and the price of accessories. Chapter 5 deals briefly with electronic typewriters and dedicated word processors. Chapters 6 – 9 form the heart of the book, Chapter 6, though on general issues to bear in mind when selecting a package in fact says very little but is followed by a clear introduction which alerts the reader to points such as cursor key legend and the difference between page oriented and document oriented systems.

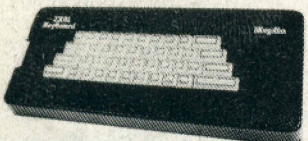
In Chapter 9 Susan Curran uses Wordstar as a yardstick by which to evaluate nine other programs. Of Wordstar, she states that the manual is poor. This in my opinion, is a gross understatement – and she also claims that it is no harder than average to learn – a comment with which I take issue. Five testpieces – covering formatting, editing, indenting, form letters and columnar work were used (where possible) with all the programs. The programs tested included Alphatext, Oliword (which I don't think can be considered "popular") and Perfect Writer, which sounds like a program I would like, but Susan Curran is not too keen on. She writes "I suspect a great many users rapidly give up and fork out for a copy of Wordstar instead". She examines Quicksilva's program for the Spectrum (her verdict on it is damning: 'nonsense to call it a word processor' and also Scripsit, Select and Spellbinder, all of which she rates fairly highly; Tasword Two for the Spectrum – with which she confesses she is "impressed" – and Wordsworth. I am very distrubed by her choice of this package as she makes the mistake of measuring all BBC Micro programs by it. She admits she is 'not familiar with' other BBC programs and it may be that some of the others are a "marked improvement on Wordsworth". Readers of *E&CM* who have seen the reviews of Wordwise and View by S. M. Gee (in July and August 1983 issues) will realise that the latter is very much a poor relation of the other two and that BBC Micro owners certainly can find a "quality" program that is at the same time relatively cheap. In short I cannot agree with the author that, when it comes to word processor packages, cheap is nasty and that the more expensive is best.

On the whole, I am more impressed with the general material than with the reviews of specific packages which I could not entirely agree with. Opinions of word processor programs are necessarily coloured by the one the reviewer is most familiar with – in Susan Curran's case Wordstar, is a program which she likes and I hate. My own preferred word processor is of the "what you can see is not what you get" variety and I felt insufficient attention was paid to the advantages of this type of program. ■

EXCITING ADDITIONS FOR YOUR HOME COMPUTER

KEYBOARD with ELECTRONICS for ZX SPECTRUM

★ Full size, full travel keyboard that simply plugs into expansion port on your Spectrum. ★ Offers single key selection of all major multi-key functions. ★ Extends port for other peripherals. ★ Can accept Atari-type joysticks (optional extra — order 2 of FG66W, £1.36 each and note that case will require cutting).
Three kits needed to build unit: Order LK29G, LK30H & XG35Q. Total price £39.95. Full construction details in Project Book 9 XA09K 70p. Also available ready-built. Order As XG36P. Price £44.95.



KEYBOARD with ELECTRONICS for ZX81

★ Full size, full travel keyboard that's easy to add to your ZX81. ★ No soldering in ZX81; simple instructions make it easy to fit. ★ Makes Shift Lock, Function & Graphics 2 single key selections.
Complete kit (excl. case) LW72P Price £23.95. Case XG17T £4.95. Full construction details in Project Book 3 XA03D. Price 70p. Ready-built in case XG22Y. Price £32.50.

MODEM

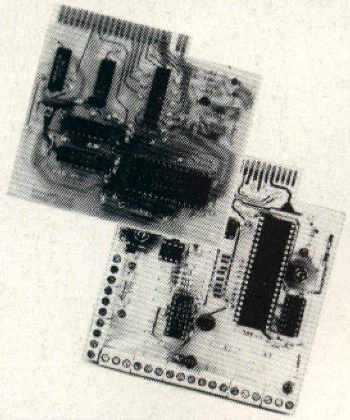
A CCITT standard modem that connects directly to your telephone line via a BT approved transformer. Transmits and receives simultaneously on European standard frequencies at 300 baud. May be used to talk to any other 300 baud European standard modem including the Maplin Computer Shopping modem on 0702 552941 and any British Telecom Datel 200/300 Service modem. The modem's computer interface is RS232 compatible.
Complete kit (excl. case) LW99H. Price £44.95. Case YK62S £9.95. Full construction details in Project Book 5 XA05F Price 70p.

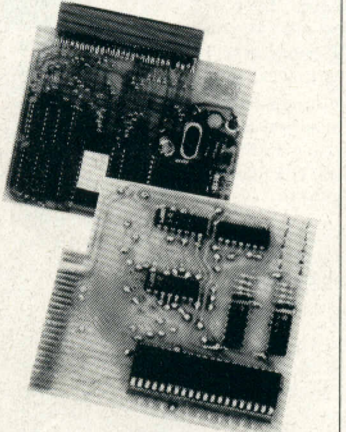
INTERFACES for MODEM

Interfaces are now available for the following machines: Commodore 64, Dragon 32, Oric, Spectrum, VIC20 and ZX81. Each is complete with a Machine Code Communications program. The BBC micro needs no interface and a suitable program is on Maplin catalogue page 15 or Project Book 8, page 59.

Computer	Order	Details	Price
64/VIC20	LK11M	Book 7	£9.45
Dragon 32	LK12N	Book 8	£13.75
Oric 1	LK40T	Book 10	£12.95
Spectrum	LK21X	Book 8	£17.95
ZX81	LK08J	Book 7	£24.95

**Project Book 7 XA07H. Price 70p.
Project Book 8 XA08J. Price 70p.
Project Book 10. XA10L. Price 70p.**





ZX81 I/O PORT

★ Provides two bi-directional ports for 16 input or output lines. ★ One buffered output which can interface directly to CMOS. ★ On board address selection permits expansion to six ports with two boards.
Complete kit LW76H. Price £9.25. Full construction details in Project Book 4 XA04E. Price 70p.

MAPLIN TALK-BACK SPEECH SYNTHESISERS

★ Unlimited vocabulary with allophone (extended phoneme) system. ★ Can be used with unexpanded Oric 1, VIC20 or ZX81 as it does not require large areas of memory. ★ Speech may be easily added to programs. ★In VIC20 version speech output is direct to TV speaker with no additional amplification needed.

Computer	Order	Details	Price
Oric 1	LK28F	Book 9	£22.95
VIC20	LK00A	Book 6	£22.95
ZX81	LK01B	Book 6	£16.95

**Project Book 6 XA06G. Price 70p.
Project Book 9 XA09K. Price 70p.**

DRAGON 32 I/O PORT

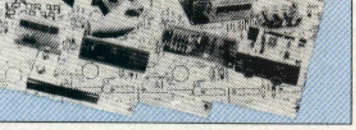
★ Provides two TTL & 3-state bus compatible 8-bit ports, ★ Four norm/inv. latched ports, ★ Two relay switched ports ★ And two opto switched ports. ★ Module plugs directly into cartridge socket and is fully programmable from BASIC.
Complete kit LK18U. Price £13.95. Full construction details in Project Book 8 XA08J. Price 70p.

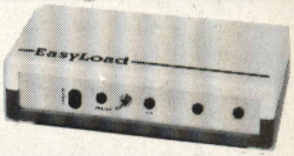
MAPLIN CATALOGUE

Full details of all Maplin's projects and electronic components in our huge 480 page catalogue. On sale now in all branches of W.H. Smith price £1.35. Or send £1.65 (incl. p&p) to our mail order address.

OTHER PROJECTS

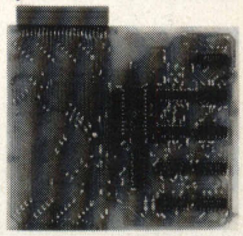
For full details of our other computer-related projects please see the relevant project book as below:
**ZX81 Sound on TV — Book 6.
ZX81 Extendi-RAM — Book 9.
VIC20 Extendiboard — Book 9.
Dragon 32 Extendiport — Book 10.
TTL/RS232 Interface — Book 9.
Project Book 6 XA06G. Price 70p.
Project Book 9 XA09K. Price 70p.
Project Book 10 XA10L. Price 70p.**





SPECTRUM EASYLOAD

★ Greatly reduces cassette LOADing & SAVEing problems on Spectrum. ★ Battery powered, no bus connections. ★ Charging from Spectrum PSU. ★ SAVE & LOAD indicators.
Complete kit (excl case) LK39N. Price £9.95. Full construction details in Project Book 10 XA10L. Price 70p.

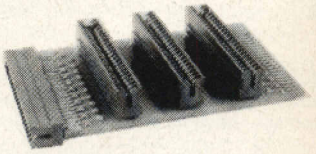


ZX81 HI-RES GRAPHICS

★ Full 256 x 192 fine pixel display with normal or inverted video. ★ Draws lines, circles and triangles, fills and textures. ★ Up to 32 user defined graphics. ★ Operates directly from extended BASIC.
Complete kit LK23A. Price £19.95. Full construction details in Project Book 9 XA09K. Price 70p.

ZX81 SOUNDS GENERATOR

★ Turns your ZX81 into a mini-synthesiser. ★ 3 programmable tone generators. ★ 3 programmable attenuators. ★ Noise generator with 3 pitch levels for special effect sounds. ★ Single address access with PEEK and POKE. ★ Connects directly to extension board or expansion port socket with extra socket (order RK35Q £1.96) ★ Requires separate amp and speaker.
Complete kit LW96E. Price £10.95. Full construction details in Project Book 5 XA05F. Price 70p.



ZX81 EXTENSION BOARD

★Plugs directly into ZX81 expansion port. ★ Accepts a 16K RAM pack and three other plug-in modules simultaneously. Parts are sold separately as follows:
PCB GB08J. Price £2.40. Edge Connectors (4 needed) RK35Q. Price £1.96 each. Track pins (1 pack needed) FL82D. Price 85p per pack of 50.

MAPLIN
ELECTRONIC SUPPLIES LTD

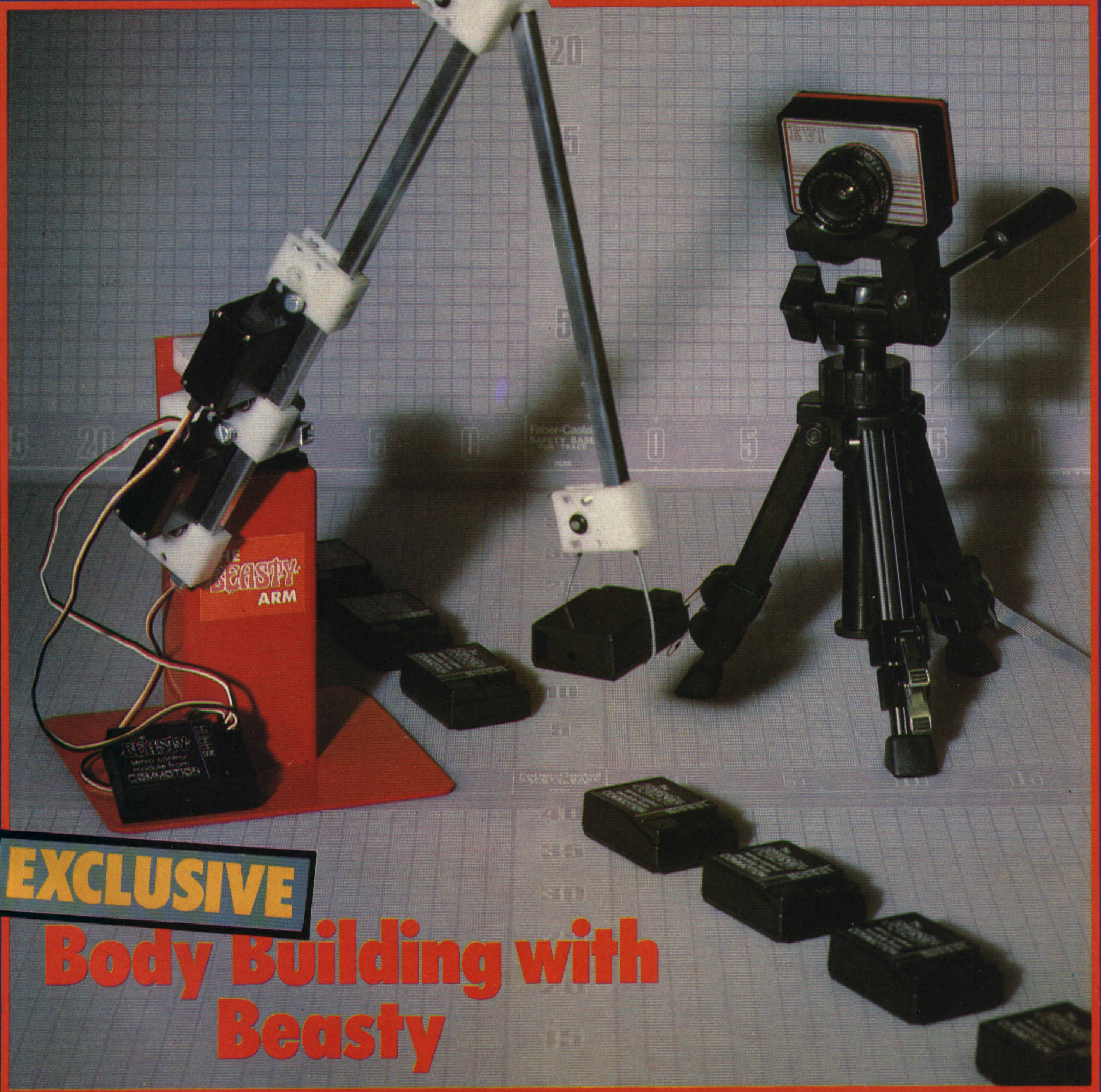
Maplin Electronic Supplies Ltd. Mail Order: P.O. Box 3, Rayleigh, Essex SS6 8LR.
Tel: Southend (0702) 552911. • Shops at: 159-161 King Street, Hammersmith, London W6. Tel: 01-748-0926.
• 8 Oxford Road, Manchester. Tel: 061-236-0281. • Lynton Square, Perry Bar, Birmingham. Tel: 021-356-7292.
• 282-284 London Road, Westcliff-on-Sea, Essex. Tel: 0702 554000. • 46-48 Bevois Valley Road, Southampton.
Tel:0703 25831. All shops closed all day Monday.
All prices include VAT and carriage. Please add 50p handling charge to orders under £5 total (except catalogue).

FREE

your ROBOT

An EMAP Publication

BRITAIN'S FIRST ROBOTICS MAGAZINE JUNE 1984



EXCLUSIVE

Body Building with Beasty

IN CONTROL WITH INTERBEEB

MOTOR SELECTION-A PRACTICAL GUIDE

EDITORIAL

VOLUME 1 ISSUE 5

WHEN IS A ROBOT NOT A ROBOT?

Last month we brought you news of the new low cost series of 'fun' robots from Prism – the Movits. Within a couple of days of the issue hitting the streets we had a slightly irrate 'phone call from one of the our advertisers saying that they thought that **Your Robot** should not concern itself with the likes of the Movits. The argument ran along the lines that as the Movits range in price from less than £10 to around £35 they could not be real robots.

This view of robots brought into question just how wide the coverage of **Your Robot** should be. After much discussion and a couple of pints of lager we thought that the decision to keep the coverage of the magazine as broad as possible is the right line to follow. The reasons for this are manifold. In the first case, the field of robotics is still in its infancy and, while we all have ideas of how it will develop over the next few years, no one can know for certain in which direction things will move. To my mind it is low cost robotic systems, in which class I would place the Movits, that will stimulate a wider interest in the field. Again the analogy with the early days of computers, while not perfect, does give an indication as to the likely trends. It was the likes of the MK14 that could hardly be called a personal computer by today's standards that introduced many people to the world of computing. In a similar fashion, calling a robot with a minimum specification such as a Movit, may be stretching things a little, there can be no doubt that products of this ilke will expose more and more people to control systems.

We'll have a fuller report on the Movits in next month's issue together with some suggestions for improving the performance of some of the family members.

THINKING CAPS ON

See page 57 of E&CM for a competition linked to the Interbeeb system reviewed in this issue of **Your Robot.**

CONTENTS

Editorial 01-833-0846
Editor Gary Evans
Assistant Editor William Owen
Production Editor Liz Gregory
Contributing Editors
Gary Herman, Peter Matthews
Advertising 01-833-0531
Advertisement Manager Richard Jansz
Production 01-833-0531
Art Editor Jeremy Webb
Make-up Time Graphics

YOUR ROBOT
2nd FLOOR
155 FARRINGDON ROAD, LONDON EC1R 3AD

NEWS

ROBOTS UNPINNED

International Resource Development Inc. reports that advances in gyroscopes and inertial positioning devices have reached a point when they will soon be incorporated in industrial robots.

When built into industrial robots, inertial guidance equipment will enable the robot to 'know' where it is at all times, to work in a much more versatile manner than the bolted-down and hard-wired robots of today. Coupled with vision systems and sensors ('collision-avoidance electronics') inertial guidance systems will allow a robot to 'go almost anywhere and do almost anything a human worker does' says the report – perhaps rather too optimistically (or is it pessimistic?).

The report also states that affordable plastic gyroscopes for automobiles will form the basis of self-navigating cars.

Errors progressively build up during the operation of any gyro based inertial guidance system. Honda are currently developing a method of periodically updating the gyro with satellite-derived positioning information. However currently available navigation satellite signals provide a new fix at a minimum of only 90 minute intervals.

ZEAKER MARK II

The Zeaker Turtle was originally commissioned and marketed by Colne Robotics. That company have now dropped the product which was, to say the least, incompatible with its range of upmarket robot arms.

Rights to the Zeaker have now fallen to its designer, David Buckley, and the product is now marketed by Intergalatic Robots (IGR). IGR intend to produce a family of robots for personal, domestic and educational use and their first step has been to upgrade the Zeaker.

Zeaker II is also a turtle type drawing robot but differs from Zeaker I in that the buffer sensors have been dispensed with and the housing vastly improved. It will be priced at under £100 (half the price of Zeaker I) and initially will be released with Spectrum Logo.

Zeaker III will be a 'buggy' type robot and sold in kit form, also for under £100. IRG can be contacted on 01 359 2536.

The other Zeaker news is that Modular Peripheral Concepts have developed a programmable joystick controller (which plugs in the expansion port of an Oric or Atmos) which can be used to control the turtle.

Spare lines in the standard joystick connector (normally used by analogue joysticks) have been connected to the remaining data lines in the controlling chip. The control unit can thus interface with any device requiring up to 16 bits of data, in or out, without special interrupts. It is thus ideal for controlling the Zeaker, and illustrates the potential of the Oric for computer control.

CNC TEACHING

Feedback Instruments have extended their range of robot and control devices with a computer numerically controlled machine tool, module CNC932.

CNC lathes and machine tools are a rapidly expanding field of industrial automation/robotics. The Feedback device is primarily designed as a training device for colleges which can't afford full scaled industrial machines.

The CNC932 is an accurate model of a co-ordinate drilling machine, equipped with electronically controlled stepper motor drive. The machine is supplied with control unit, power supply, and instruction manual with lots of program listings.

The equipment connects to a range of popular micros via the Feedback MICA interface, but software is currently available only for the BBC micro. Feedback Instruments Ltd., 08926 3322.

The Beasty GROWS

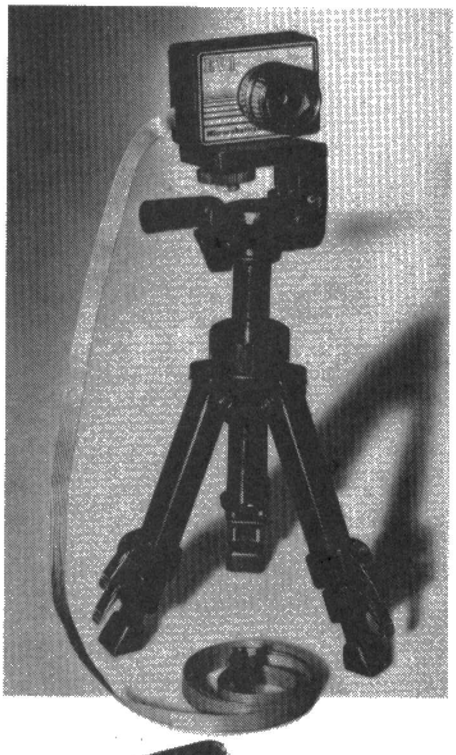
A look at Commotion's new arm that complements their Beasty servo controller and the companion vision system.

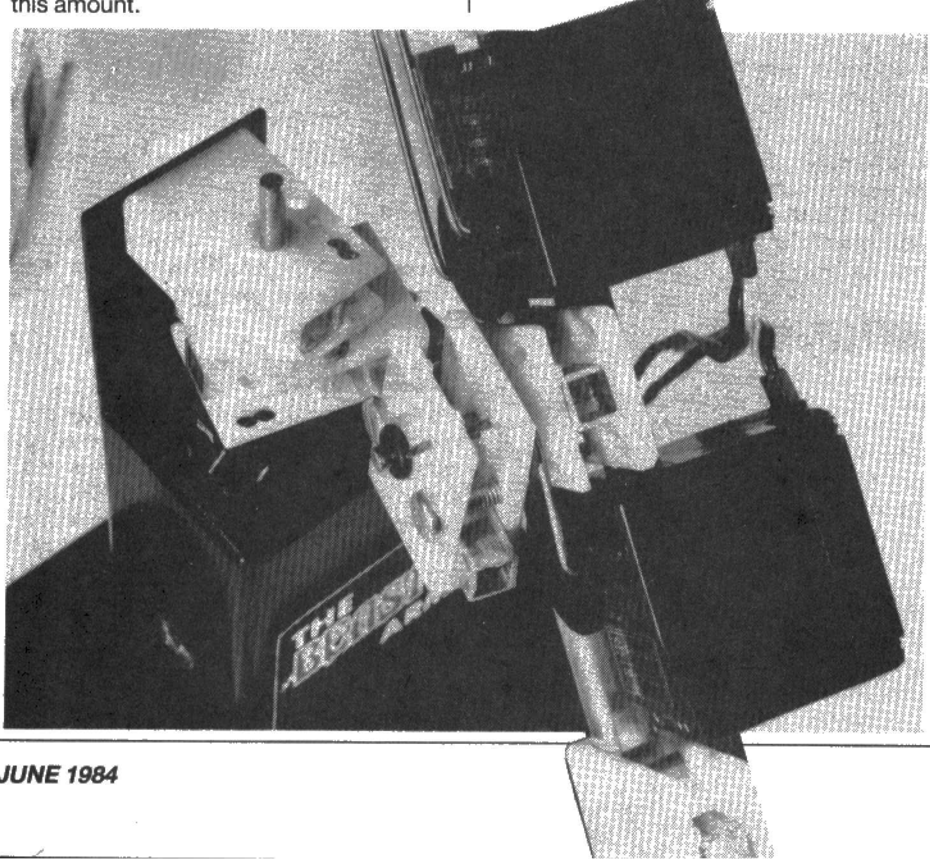
The arm, called 'Snap', consists of a steel base, 9″ aluminium arms and moulded joints. It is driven by 3 standard Beasty servos and connects to a BBC Model B using a Beasty. Many people will already have suitable servos to drive the arm since they are made by Futaba and are used in radio controlled models.

The arm is easily assembled using a small screwdriver (screws are used to clamp the moulded joints onto the arm sections). It is possible to experiment with building different sized robots by using different lengths of ¼″ aluminium section.

The Beasty can be programmed to pick things up using the hook on the end of the arm. Alternatively, by adding a further moulding and servo, a gripper can be made.

The arm is very fast, capable of moving over its entire range in less than a second (however, anything it is holding will go flying unless it is restrained a little!). The 3 axes move through a 100 degree range giving it a slightly smaller working range than its rivals. Using standard servos the arm can lift 70 grammes; more powerful servos can be used allowing it to lift twice this amount.





The BEASTY arm's low price (£39.95, but the servos are extra) and flexibility make it ideal for the experimenter, and we will be featuring it in future articles. It does not have the lifting power of larger robots but makes up for it with speed.

SNAP

Launched at the same time as the arm is an electronic vision system called EV-1 (also known as Snap).

The camera looks like a spoof, housed in a small 4″ by 3″ box. It has a Pentax 110 lens mounted on the front and an 8-way ribbon cable coming out of the side. It is not a spoof at all; it is a complete computer vision system which plugs into the BBC Model B userport and produces images on the screen when the software is loaded. EV-1 costs £129.95 inc. VAT.

There are a number of things to consider before buying a vision system: how fast is it; what resolution is it; what lens attachments can be used; can it be mounted on a robot arm?

Snap can run at 20 frames a second, has a resolution of 128 by 256, takes interchangeable camera lenses and, with a weight of 45g, can be mounted anywhere!

It works well especially when looking at black objects on a white background, it does unfortunately have a discontinuity half way up the sensor, this creates a 'blind spot' of about 8 pixels, and is caused by a power line running through the DRAM used as the imager.

Applications abound: recognising industrial parts, counting, reading text, tracking movement. The camera comes with a large quantity of software, covering imaging, outlining movement detection, object location, area calculation, and object recognition. Much of the software is listable, and is commented to aid understanding and to help those interested in developing the software further.

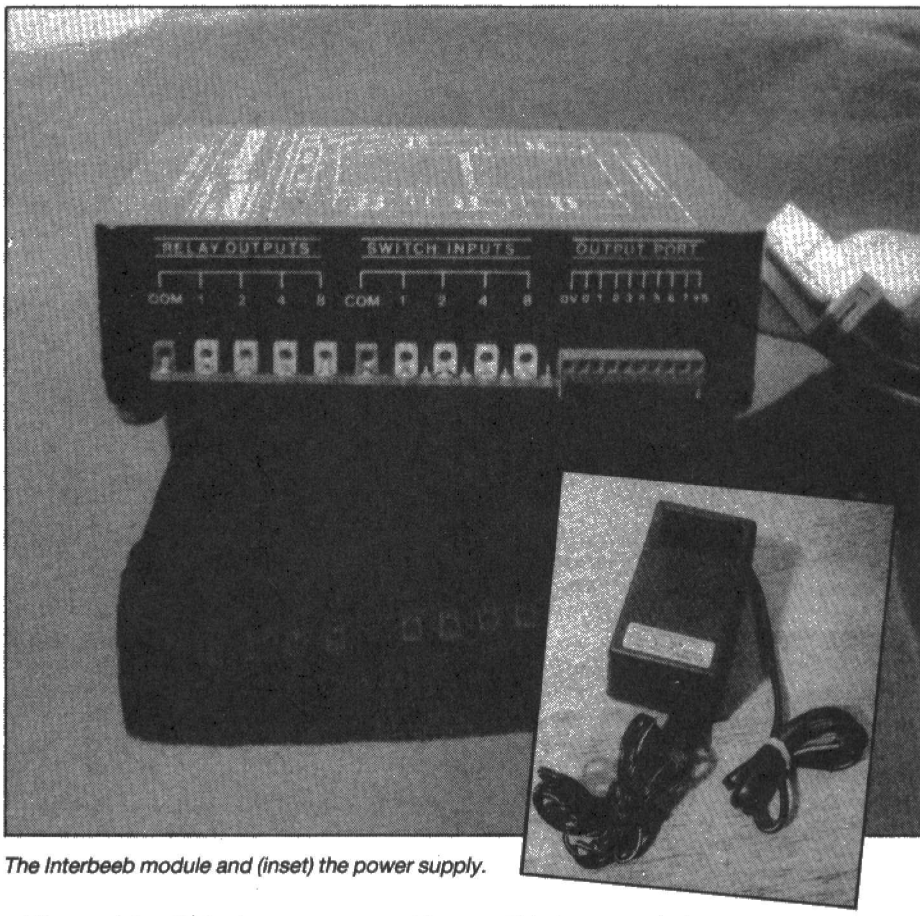
The software has been developed on the BBC Model B, but the camera hardware can be attached to any computer that has a 6522 port available or, using their parallel interface card, to virtually any computer with a parallel port. Commotion say they will convert the software package for any computers for which sufficient interest is shown. ■

INS AND OUTS OF INTERBEEB

DCP Microdevelopments Interbeeb is a versatile interface package that extends the I/O capabilities of the BBC micro. Paul Hickling puts the unit through its paces.

The BBC micro is well endowed with I/O facilities yet, for control applications, even this computer lacks the necessary range of output drivers and input channels. In addition, in many cases it would be unwise to connect equipment which may be of an experimental nature direct to the BBC's bus. This is because of the somewhat fragile, in the electrical sense, nature of the BBC micro's I/O ports. Some form of buffering between the micro and the equipment which it is controlling is essential if damage to the computer is to be avoided.

The Interbeeb is designed to fulfil both of these requirements, providing a range of output drivers, including a high current relay driver, enhanced input facilities, and an eight channel analogue to digital converter which takes pride of place here, plus full buffering of the computer.



The Interbeeb module and (inset) the power supply.

HOOKING UP

Interbeeb is supplied with a separate power supply and so makes no demands on the internal supply of the micro. This is connected to Interbeeb via a 3.5mm jack plug while the unit is connected to the micro's 1MHz bus via an IDC connector. With these two connections made the Interbeeb is ready for use.

INPUT AND OUTPUT PORTS

Interbeeb provides two 8-bit TTL ports for input and output. Both ports are mapped to address &FCC", with the input function being called by preceeding this location with a function (ie PRINT) and output initiated by following the address with a value.

Connections to both the input and the output are made via 10-way Molex sockets. The manual gives some basic examples of connections to these ports in the form of the circuitry necessary to hook up a series of switches to the input port (together with details of the pull up/down resistors) and for driving a series of LEDs from the output port.

The most obvious application of these Interbeeb ports is the connection of other TTL based equipment to the BBC. This is because both ports impose the usual constraints when using TTL devices, namely that inputs must be in the range 0 to 5V and that outputs lie in the same voltage range while providing limited output current (the Interbeeb's output lines are capable of sinking 15mA per channel). The unit features additional ports that are free from these limitations.

SWITCH INPUT

These four inputs are similar to the TTL inputs, the difference lying in the fact that the lines are pulled down by internal 2k2 resistors. Connection to these inputs is via 2mm sockets providing a simple yet reliable electrical connection.

This port is read in the same way as the TTL input port although it is located at a different address: &FCC1. The inputs of this switch input correspond to lines D0 to D3 on the data bus while bits D4 to D7 are held low by the Interbeeb. Reading the switch input will thus result in a number in the range 0 to 15.

This input is suitable for inputting information from transducers including light dependent resistors, microswitches, and pressure switches. The transducer is wired between the appropriate switch input and a common line also brought out on a 2mm connector.

RELAY OUTPUTS

The four relay outputs allow the BBC micro to control high current (up to 1 Amp at 12 volts) devices including motor, lamps and solenoids. The output is called at &FCC1, the same address as the switch port with a call to this port being differentiated from one to the switch port in much the same way as was the case in the TTL port.

ADC

The ADC is located at &FCC) and provides 8 inputs capable of converting an analogue

voltage in the range to 2.45 volts. The Interbeeb makes available a precision 2.45 volt reference source making connection to resistive transducers, such as the potentiometers of a joystick, quite straightforward.

When in use, the channel to be converted must be specified by using the command ?&FCC-n where n is number between 1 and 7 corresponding to the channel to be converted. This operation starts the conversion and as conversion time is about 100us, when using BASIC conversion is complete by the time the channel is read with a PRINT ?&FCC).

The manual provides a simple joystick program which can be used to demonstrate the action of two of the ADC channels.

BUS SYSTEM

In addition to the inputs and outputs described above the Interbeeb features the DCP bus system, something that is common to other products in the Company's range. The bus uses a 15-way 2.54mm pitch socket and provides access to the computer's data bus, RD, WR signals and +5V and +9V power supplies. In addition the bus provides two lines, AD1 and AD2 which are fully decoded at addresses &FCC3 and FCC4 and gated with the BBC micro's valid address signal.

THE LAST WORD

The DCP Interbeeb is a well constructed unit that provides a full range of buffered I/O functions that should cater for most of the demands of a computer control environment. The Company's range of products include a fast AD pack (conversion time less than 10uS) and a precision digital to analogue conversion pack. Both these products are used in conjunction with the DCP expansion bus.

As well as the above, DCP also produce Interspec which, as the name suggests, is a device offering similar features to those of the Interbeeb but designed for use with the Spectrum computer.

Interbeeb retails for £69.95. The DAC pack for £19.95 and the ADC unit also for £19.95. Interspec costs £49.95 and all prices include VAT. The DCP range of products is available from many dealers or direct from RH Electronics (Sales), Chesterton Mill, French's Road, Cambridge.

Motors explained

In the first of a series of articles, D. S. King* looks at the various types of motors available to the robot builder.

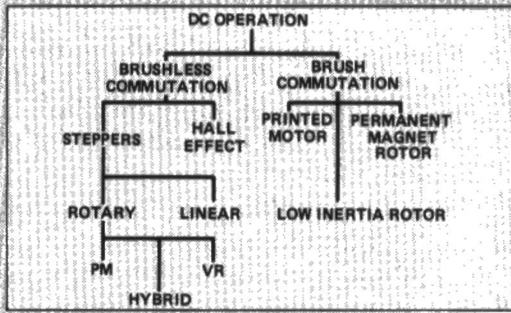


Figure 1. DC motor's family tree.

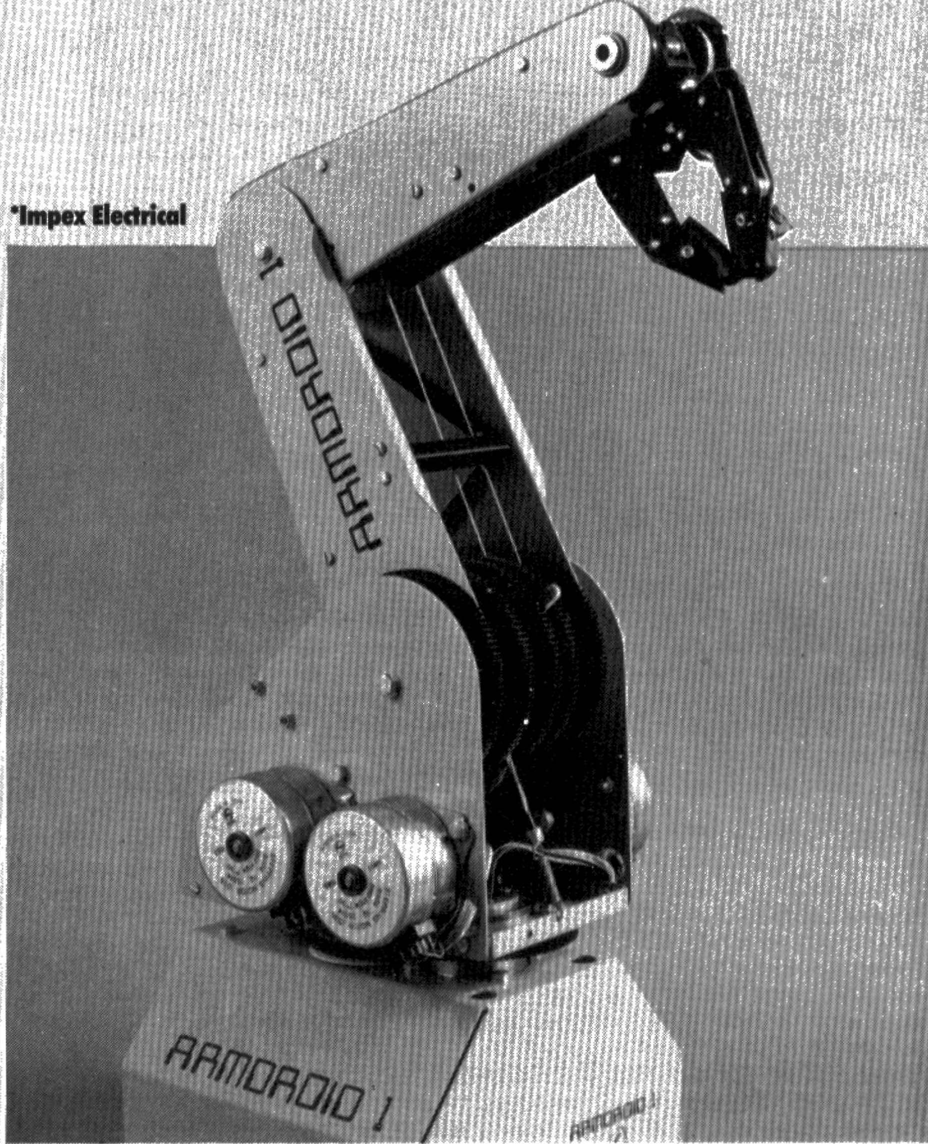
When contemplating the design of any motorised system it is essential that due care is exercised when selecting the type of motor to be employed. But whether a DC motor in a closed loop is preferable depends on achieving a close match between motor and load inertia. And don't ignore the motor controller for it could be a single IC that puts the motor through its paces.

On the robotics side we have seen recently the emergence of robotic arms such as the Colne Robotics' Armdroid, see photo, the Edinburgh Turtle, and Economatics' BBC Buggy. In the field of instrumentation, where it is used for computer data presentation, there is equipment like Parfit Electronics' X-Y plotter that uses stepper motors.

For each application it is small, low powered motors which provide either a transmission drive or some form of positioning function. The motor may be a stepper or one of the various designs of DC motors, **Figure 1.**

A major influence on motor selection for computer associated equipment is the microprocessor. Because of its compatability with computer signals, the stepper will almost be considered as first choice simply because of its digital nature. But even though it may also be the first choice from a cost aspect, this type of motor may not be always the best advice for a specific type of load moving application.

***Impex Electrical**



Colne Robotics' Armdroid employs five stepper motors.

SINGLE CHIP CONTROL

The controller which feeds signals to the motor also needs careful consideration.

> **". . . selecting the right motor for the job is essential . . ."**

Electronic circuit design has advanced to the stage where an integrated circuit (IC) can provide all the necessary operational capability wrapped up in a single chip. In fact some IC controllers can be operated via an ASCII keyboard and interfaced to an 8-bit processor bus. By means of a host processor the stepper can, for example, be made to take a single step, accelerate/decelerate, or run at constant speed.

These monolithic ICs are capable of bridging the gap between low level signals and high current motor loads. It is an approach that results in a less complex controller with reduced bulk, thus leading towards higher levels of reliability and greater cost effectiveness.

ICs that are available for the control of small motors range from single arrays of drivers to microcomputer-based 'intelligent' controllers. They can accommodate low level logic signals or pulse trains, handle analogue feedback and sensing information, such as the kind that is delivered from servo potentiometers, optical encoders or tachogenerators.

There are IC control devices that implement closed-loop servo-positioning systems. When under software control, the chip set renders high accuracy positioning and, at the same time, is capable of producing up to 7A at 36V for driving dc motors. With this type of IC, dual servo modes are automatically implemented. Initially, a velocity-control mode drives the motor towards the 'target' position. As this final position is approached a position control mode electronically halts the motor precisely at the desired point.

Some IC controllers consist of up to three chips. A set of this type offers a different approach. Functions that often require adjustment or ramp slope are performed in the analogue domain, therefore no costly software programming is needed. Also available are CMOS bipolar digital/linear chips that will accept logic inputs for pulsing steppers bi-directionally.

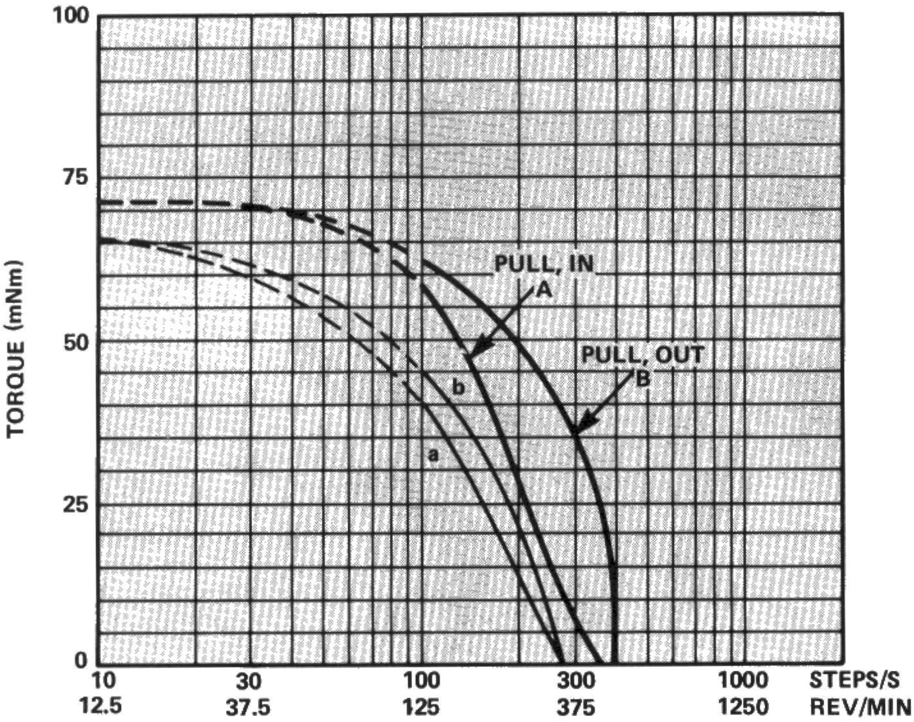


Figure 2. The torque vs stepping rate of the motors used in Coln's Armdroid.

Then there are circuits that operate pulse-width-modulated (PWM) signals, a technique that gives high efficiency control to dc motors.

Although it is possible to collect together a variety of discrete and monolithic components, such as power transistors, comparators, and OP-amps, for building effective cointrol circuits, some ICs are designed especially for motor control.

One of the latest to come on to the market is Mullard's SAA-1027 dual in-line, 16-lead device for driving 4-phase 2-stator steppers. It operates off a supply of between 3.5 and 18V (4.5mA unloaded). It features a high output current of 500mA with the ability to drive a motor clockwise or anti-clockwise; its outputs are protected against motor overshoot.

"...effective control of motors can be accomplished by discrete circuits but there are ICs designed for the task..."

PERIPHERAL USES

Clearly, there is a wide choice of devices for controlling small motors, but it still doesn't make the selection of a motor any easier. The drive requirements of different computer peripherals highlights the necessity to know which motor best suits which type of application.

Amongst the main areas of application for dc motors and steppers are computer data printers which generate the hard copy necessary to pass information from one individual to another; and auxiliary storage devices that have to store large quantities of data and deliver it to the computer as, and when, required. If the machine is a line printer it will print a whole line of characters simultaneously at speeds typically in the region of 300 to 2000 lines/min. On the other hand, the serial printer, which is for low cost, high volume applications, prints out its characters one at a time in the same way as an electric typewriter.

Some printers employ both dc motors and a stepper and, in certain applications, they compete.

A printer may utilise motors by having dc servomotors for driving the carriage and print wheel, and steppers for the ribbon drive and paper feed. The actual printing function is often performed by a solenoid.

Servomotors are chosen to drive the print wheel and carriage because they can be halted abruptly without oscillation, thus permitting clear character printing. Stepper motors are used because of their relative simplicity and low cost.

For the fast growing application of floppy discs, disc pack and disc cartridge, both types of motor can be used. However, for peripherals which have reel-to-reel magnetic tape drive, dc motors are preferred for capstans and cassettes.

ROBOTICS APPLICATIONS

The Colne arm referred to above illustrates a typical example of the use of steppers in robotics. The requirement for this application is to have an arm with 5-axes of rotation plus a 3-finger gripper.

The drive needs to be programmable as a continuous path – which uses several axes simultaneously – It has to function with any commercially available microcomputer with an 8-bit parallel port. The balance of operational requirements favours high working and holding torques, while relatively low response times are acceptable. **Figure 2** shows the characteristic torque vs stepping rate for the steppers used in the arm. Also, since the target market includes educational and home enthusiast users, stepper motors which operate directly in open-loop are used for economic reasons.

One geared motor is located in the base as a turret drive (360°), and five geared motors located in this turret, drive the other axes via a pulley system.

Permanent-magnet-rotor types are used as the reduction gearing increases the torque for indefinite stationary holding position. A reach of 430mm with a load capacity of 300g is achieved with a positional resolution of 4mm.

These steppers are 4-phase uni-polar devices. Alternatively, DC motors that are used for low power, computer controlled applications may have either a mechanical or brushless commutation system. The main differences between the way a stepper and a DC motor operates is that the former device works on open loop whilst the dc motor performs in a closed loop. This difference affects the way in which a motor is controlled, as indicated in **Table 1.**

Continued next month.

TABLE 1. Motor system requirements.

Steppers	Open loop, that is input information determines motor position and speed.
D.C. motors	Closed-loop, that is feedback required for:
1.	Positional information from shaft encoder which supplies absolute or incremental change data in BCD form.
2.	Speed information which can be derived from:
a)	Back EMF reading in PM motors. Limited use in computer linked applications. Mainly used in constant speed applications, such as audio and video equipment.
b)	A.C. tacho outputs which can be as speed/voltage or speed/frequency related. The latter method is more accurate, but control circuitry expensive. No directional information.
c)	D.C. tacho outputs which give instantaneous speed and direction information. More accurate that a.c. tacho, especially at low speeds.